

LAUNCHER Office

LAUNCHER Office est la solution dédiée à la gestion des flux de documents d'entreprise issus de vos systèmes, applications et SGBD, développée par Aura Equipements.



Smart Software for Powerful Business

Elle vous permet de définir, concevoir et piloter des processus complets de gestion des flux de documents.

Concrètement, **LAUNCHER Office** pilote pour l'utilisateur, l'application ou le programmeur :

- Les applications Microsoft Office (Word et Excel)
- L'envoi de mail (à partir d'Outlook, de Lotus ou en SMTP)
- La génération de fichiers (au format PDF, par exemple)
- Tout autre programme Windows

L'utilisateur dispose notamment de ces fonctions :

- Lancement d'un programme sur le PC avec envoi de paramètres et attente éventuelle d'un résultat ou de la fin d'exécution,
- Extraction des informations de bases de données ou d'applications distantes,
- Lancement de Word ou Excel, chargement de documents ou modèles,
- Affichage de menus personnalisés,
- Modification de documents, publipostage,
- Sauvegarde des documents modifiés, impression,
- Conversion en format PDF,
- Composition et émission de messages électroniques, avec pièces jointes,
- Transfert de fichiers ou de requête vers un classeur Excel, ou une base de données ACCESS.

Ces fonctionnalités sont mises à disposition sous forme :

- D'un ensemble de fonctions en Java sous Windows et Linux
- De webservices REST appelables par tout langage sous Windows et Linux



- De commandes CL sous AS/400 ;

Voici une liste non exhaustive de commandes :

- Créer des lettres Word en temps réel ([LNCPRTDOC](#)),
- Composer des classeurs Excel ([LNCTOxls](#)),
- Afficher des documents Word, PDF, Excel, HTML ([LNCSHELL](#)),
- Envoyer les documents générés par Fax, email ([Envoi de Mails](#), [LNCSNDMAIL](#)),
- Transférer les données issues de requêtes sur les bases de données vers les utilisateurs PC ([DBXFER](#)),
- Remplacer des éditions Spools ([Envoyer un spool](#)),

Voir : [Liste des commandes](#) incluses dans le produit LAUNCHER Office.

Composition du Produit

LAUNCHER Office est constitué :

- **Du serveur LAUNCHER**, destiné à recevoir des requêtes des clients LAUNCHER.
Ce serveur inclut des connecteurs en entrée (par exemple : AS/400 et/ou SGBD) et en sortie (par exemple : Word et/ou PDF), en fonction de la licence que vous avez acquise.

- **De clients LAUNCHER**, utilisés par des applications ou programmes sur Windows, Linux ou AS/400, qui soumettent des demandes au serveur LAUNCHER.
Les clients LAUNCHER sont constitués, en fonction de la licence que vous avez acquise :
 - o D'un ensemble de classes Java pour les clients Windows et Linux
 - o D'un ensemble de commandes CL pour les clients AS/400 (une bibliothèque d'objets nommée LAUNCHER),
 - o D'un serveur de webservices REST pour les clients Windows et Linux.
Tout langage de programmation est susceptible de pouvoir appeler ces webservices.

Fichiers et Répertoires d'installation

La partie serveur de LAUNCHER Office doit être installée sur chaque machine Windows destinée à recevoir des commandes provenant venant des programmes Windows, Linux ou AS/400.

Le répertoire d'installation se nomme par défaut :
« C:\Program Files\LAUNCHER<X> »

(par exemple C:\Program Files\LAUNCHER400 ou
C:\Program Files (x86)\LAUNCHER XPRESS).

Ce répertoire est indépendant.

Le programme LAUNCHER doit être démarré sur la machine d'installation, afin de lancer un processus « Deamon », en attente de requêtes venant des programmes Windows, Linux ou AS/400.

Ce lancement peut être automatisé au travers du groupe de démarrage de Windows, ou bien en utilisant les services Windows.

Une icône apparaît sur la barre de tâches (si vous n'utilisez pas le mode service Windows).

Configuration Requisite

PC/Serveur sous Windows

- Systèmes d'exploitation : toutes les versions supportées par Microsoft.
- Protocole TCP/IP.

Logiciels :

- Microsoft Office (toutes les versions supportées par Microsoft).
- Optionnellement : messagerie compatible MAPI (Outlook, Exchange), Lotus Notes ou connexion à un serveur SMTP.
- Optionnellement : drivers JDBC et SAP JCO

Si utilisation d'un client AS/400

- Tous les modèles AS/400 depuis la série B.
- OS/400 depuis la version V5R2.
- Connexion entre les PC et les AS/400 par le Protocole TCP/IP.

Si utilisation des webservices

- JRE8 (Java SE Environment 8) installé sur le PC/Serveur Windows

Installation

Installation de LAUNCHER Office

L'installation de **LAUNCHER Office** s'effectue à partir d'un PC et se décompose en :

- L'installation de la partie serveur, des exemples, et éventuellement des classes Java et du serveur de webservices côté Windows
- Eventuellement l'installation de la partie cliente sur l'AS/400 (bibliothèque LAUNCHER).
La partie cliente n'est à installer qu'une seule fois sur l'AS/400.

Lancer le programme d'installation "en tant qu'administrateur" est recommandé.

Conditions préalables

Sur le PC Windows :

TCP/IP fait partie des protocoles de base installés sur une station Windows.

Aucun autre logiciel n'est requis pour connecter le serveur et les clients LAUNCHER.

Sur l'AS/400 :

Le poste de travail doit être connecté à l'AS/400 par le biais d'un **lien TCP/IP**. Sur l'AS/400, TCP/IP doit être installé, configuré et démarré.

Le serveur FTP n'est utilisé par LAUNCHER Office que pour l'installation de la partie cliente sur l'AS/400. Pour l'exploitation normale des applications, il peut être arrêté.

Pour l'installation, le profil utilisateur de l'AS/400 à utiliser doit être du type **'QSECOFR'** (il est possible de l'indiquer lors de l'installation).

Pour installer la partie cliente, il faut donc connaître le nom de l'AS/4000 **ou** son adresse IP et un profil de type QSECOFR et son mot de passe.

Certaines applications de sécurité peuvent gêner l'installation de LAUNCHER Office en bloquant l'importation de données externes. Si c'était le cas, désactivez, le temps de l'installation, l'application en question.

Procédure à suivre

Exécutez le programme d'installation "en tant qu'administrateur" (recommandé).

- Renseignez les boîtes de dialogue et suivez les instructions.
- **Si vous avez demandé l'installation des programmes AS/400**, renseignez la boîte de dialogue relative à cette installation.

Pendant l'installation les opérations suivantes seront effectuées :

Sur le PC :

- Création d'un répertoire « Program Files \LAUNCHER<X> » (par exemple C:\Program Files\LAUNCHER400),
- Création d'un sous répertoire spécifique pour les exemples,
- Copie de divers fichiers.

Sur l'AS/400 :

- Création d'une bibliothèque **LAUNCHER**,
- Création de divers objets dans cette bibliothèque.

Après validation, l'installation lance le transfert des programmes sur l'AS/400.

En cas d'échec, vous pouvez relancer le transfert des programmes AS/400 par le programme "Installation AS/400" à partir du groupe de programmes "**LAUNCHER400**" qui a été créé.

La durée d'installation des programmes AS/400 peut varier sensiblement (de 5 à 30 minutes) en fonction du type et de la configuration de votre AS/400.

Installation

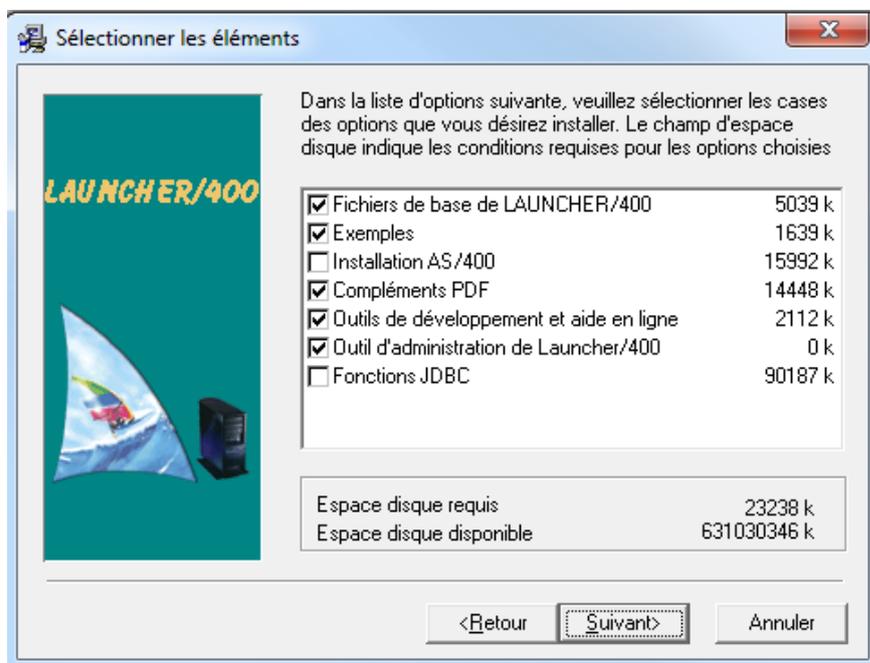
Suivant la licence choisie, Launcher Office se décline en différents produits :

- **Launcher/400** : spécialement dédié aux clients AS400
- **Launcher XPRESS** : pour toute application/programme client sur Windows ou Linux, dans n'importe quel langage.

Launcher/400

Si vous installez le produit LAUNCHER400, par défaut vous installerez uniquement la partie serveur côté Windows.

Avec le même exécutable d'installation vous pouvez choisir d'installer la partie cliente côté AS400 (en cochant « Installation AS/400 »).



Launcher XPRESS

Si vous installez le produit Launcher XPRESS, alors vous installerez par défaut le serveur Launcher côté Windows ainsi que le serveur de webservices :

 Sélectionner les éléments ✕

Dans la liste d'options suivante, veuillez sélectionner les cases des options que vous désirez installer. Le champ d'espace disque indique les conditions requises pour les options choisies

| | | |
|---|---------|---|
| <input checked="" type="checkbox"/> Fichiers de base LAUNCHER/i | 19261 k | ▲ |
| <input checked="" type="checkbox"/> Exemples | 1573 k | |
| <input type="checkbox"/> Réservé | 0 k | |
| <input checked="" type="checkbox"/> Complément PDF | 24840 k | |
| <input checked="" type="checkbox"/> Aide en ligne | 3176 k | |
| <input checked="" type="checkbox"/> Administration LAUNCHER/i | 0 k | |
| <input checked="" type="checkbox"/> Connecteurs | 99993 k | |
| <input type="checkbox"/> SAP Gateway | 467 k | |
| <input checked="" type="checkbox"/> REST Gateway | 51498 k | ▼ |

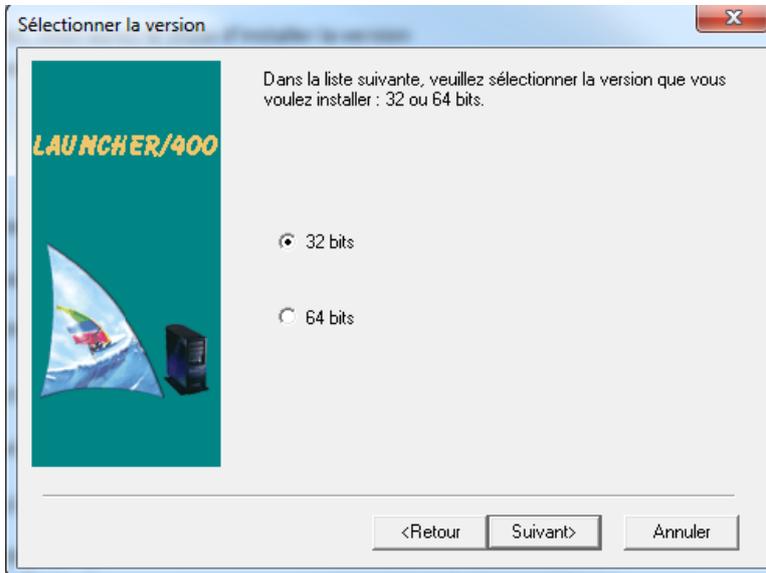
Espace disque requis 201012 k

Espace disque disponible 1808206268 k

<Retour
Suivant>
Annuler

Version 32 ou 64 bits

Si vous lancez l'installation sur un PC/Serveur Windows 64 bits, vous aurez le choix d'installer la version traditionnelle 32 bits ou bien la version 64 bits du serveur Launcher/400 :



Il est conseillé de choisir la même version que Microsoft Office.

Si vous avez Microsoft Office 32 bits, alors installez la version 32 bits de Launcher Office.

Si vous avez Microsoft Office 64 bits, alors installez la version 64 bits de Launcher Office.

Par ailleurs, si vous utilisez l'envoi de mail avec Lotus Notes, alors il faut absolument installer la version 32 bits de Launcher Office.

Licence produit

Launcher 400

Si vous utilisez le produit LAUNCHER400, les licences sont installées sur l'AS400.

Vous obtiendrez ces licences par email du service commercial.

La licence est propre à chaque AS400 et est indépendante des PC/Serveurs Windows.

Clé d'activation AS400

La clé d'activation AS400 est nécessaire si vous voulez utiliser un client AS400.

La clé d'activation est notamment calculée à partir du *numéro de série de l'AS/400*.

Pour connaître ce numéro vous devez taper sur un terminal la commande :

DSPSYSVAL SYSVAL(QSRLNBR)

Dés que vous recevez votre licence, vous devez saisir le code d'activation pour votre AS/400.

Sur un terminal AS/400, faire :

**CHGCURLIB LAUNCHER
EASYREG <F4>**

Saisir les paramètres de clé communiqués par Aura Equipements :

| | |
|----------------------------------|---|
| License | LNC |
| Special Development License . . | *YES |
| Company name | Nom_de_votre_Société |
| Activation Key | Votre_clé |
| Number of connections | selon la licence |
| Authorized partition ID (1->n) | 0 |
| EASYCOM Version | 3 |
| EASYCOM Options | 0 |
| Expiration date (ddmmyyyy) . . . | 00000000 |
| Authorised Proc.group | P05 or P10 or P20 or P30 or P40 or P50 |
| Extended license | *NONE |

Attention : Différencier O (lettre) de 0 (chiffre) et I (lettre) de 1 (chiffre).



Il est conseillé d'arrêter les connexions LAUNCHER avant de saisir la (les) clé(s).

Launcher XPRESS

La licence est propre à chaque PC/Serveur Windows sur lequel est installé le produit.

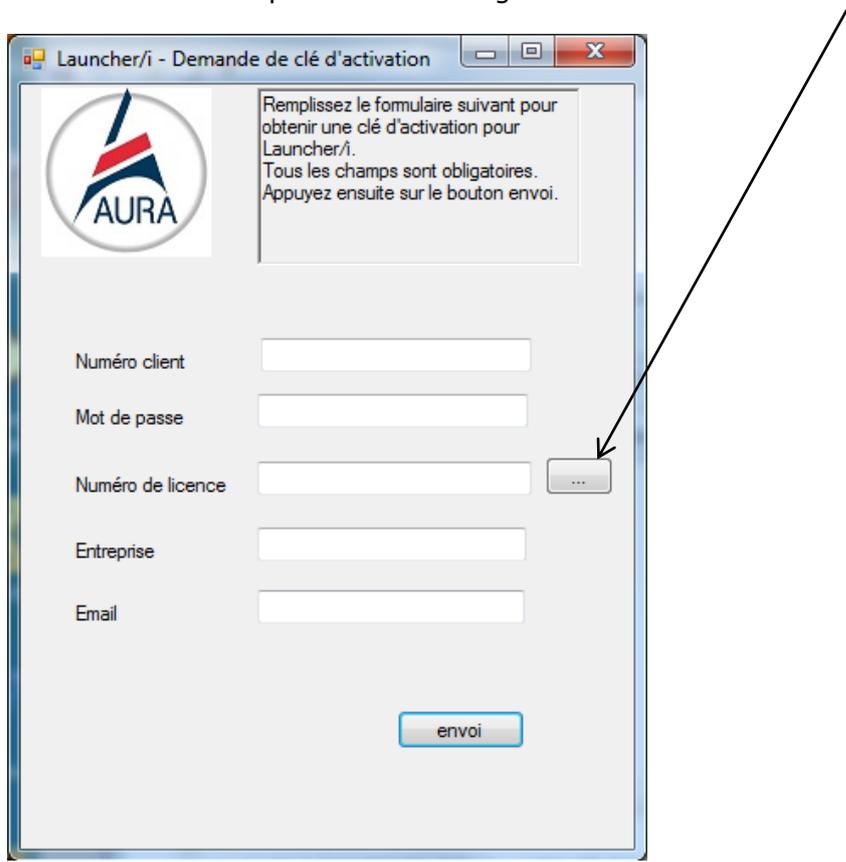
Veillez noter : 1 numéro de série du produit Launcher = 1 seule clé d'activation pour un seul serveur Launcher

Pour utiliser Launcher XPRESS il faut enregistrer une clé d'activation sur le PC hébergeant le serveur Launcher.

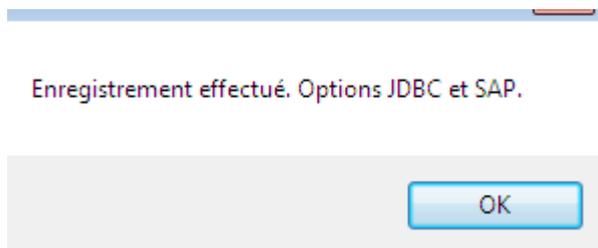
Pour obtenir la clé d'activation et l'enregistrer sur votre PC/serveur, il faut exécuter le programme suivant avec des droits d'administrateur :

C:\Program Files (x86)\LAUNCHER<X>\Licence\registerKey_FR.exe

Pour connaître les numéros de série des produits Launcher en votre possession, utilisez ce bouton après avoir renseigné votre numéro de client.



Remplir tous les champs et appuyer sur envoi. Une fenêtre pop-up apparaît :



Vous recevrez alors un email récapitulatif :

Vos références :

Numéro client : 111111

Société licenciée : company

Madame/Monsieur,

*Pour la licence Launcher/i N° AURA-PONTPY058-NS, le code d'activation est :
55109ADDE1232503728968.*

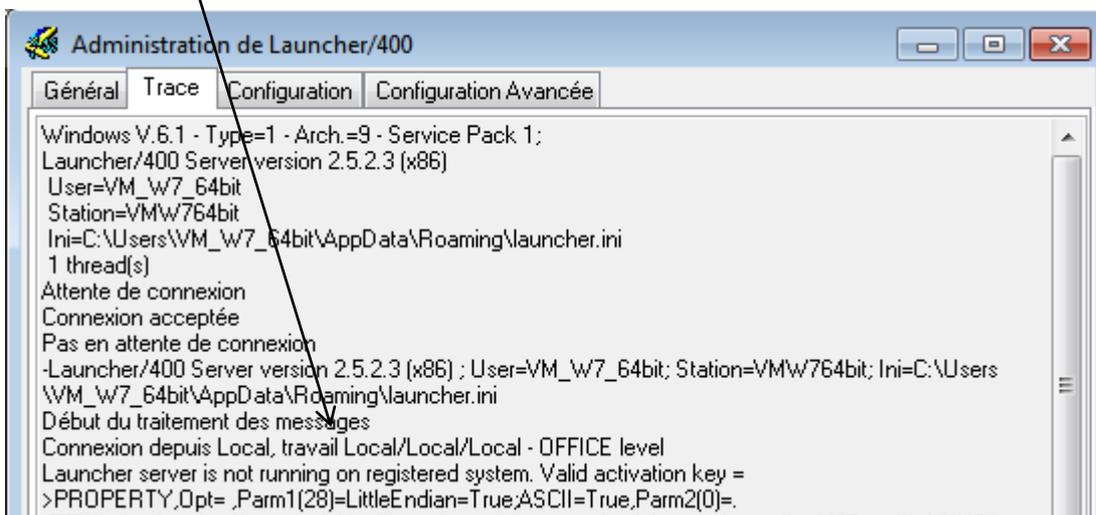
Cette clé a été enregistrée sur votre PC hébergeant le serveur Launcher/i.

En outre, vous avez souscrit aux options SAP and JDBC.

Cordialement,

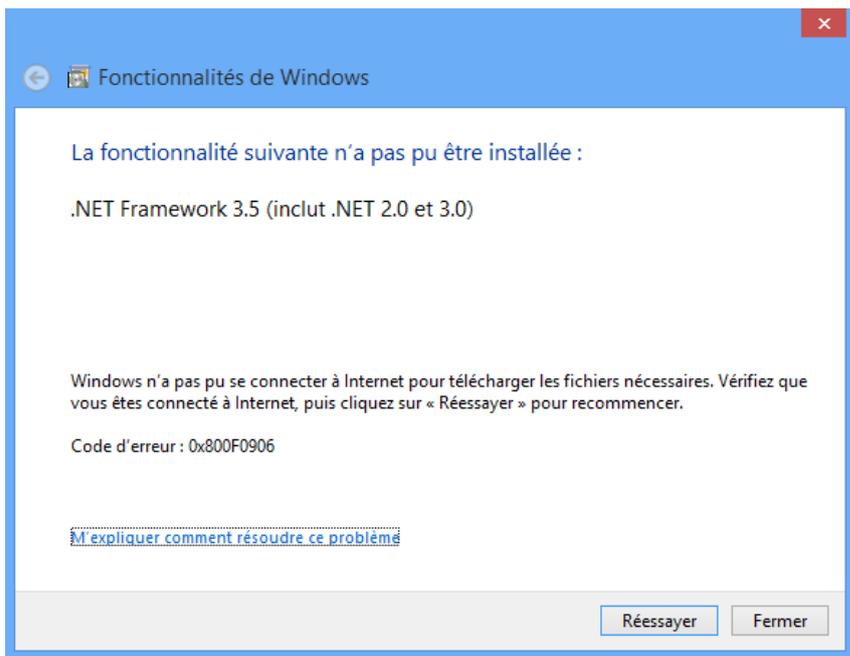
AURA Equipements

Si vous utilisez le client Launcher (Java ou webservices) avec un serveur Launcher qui ne possède pas de clé d'activation, voici le message que vous aurez :



Prérequis à l'exécution du programme C:\Program Files (x86)\LAUNCHER<X>\Licence\registerKey_FR.exe :

Sur certains systèmes tels que Windows 8, le Framework 3.5 (inclut .NET2.0 et 3.0) peut ne pas être présent, et vous aurez ce message lors du lancement du programme :



Procédez au téléchargement en utilisant le lien suivant et procédez à l'installation :

<http://www.microsoft.com/fr-fr/download/details.aspx?id=21>

Installation sur une station, en mode silencieux

Il est possible d'avoir un programme d'installation automatique (LNC_xxxxAUTO.EXE).

A demander au service technique d'Aura (tech@easycom-aura.com) si vous avez un contrat de maintenance valide.

Ce programme installe la partie Windows de LAUNCHER Office, sans intervention d'un opérateur.

Des paramètres peuvent être passés lors du lancement de ce programme :

/s : Active le mode "silent", c'est-à-dire sans aucun affichage.

/invisible : Met automatiquement l'option pour que LAUNCHER Office démarre en mode invisible

/nopdf : Ne pas installer les composants PDF.

/withadmin : Installe l'outil d'administration de LAUNCHER Office.

/noautostart : Ne pas mettre LAUNCHER Office dans l'auto-démarrage.

/appdir="répertoire d'installation" : Change le répertoire d'installation de LAUNCHER.

/64 : Installation de la version 64 bits de Launcher.

/nostart : Launcher ne démarre pas à la fin de l'installation.

Si seule l'option **/s** est précisée, l'installation se fait selon les options suivantes :

- Installation des fichiers de base et support PDF (sont exclus : les exemples, l'outil d'administration...)
- Mise en place de l'icône dans l'auto-démarrage

Installation en mode SERVICE Windows

Généralités

L'installation de LAUNCHER Office en mode "Service Windows" permet de le faire fonctionner sur un serveur dédié sans qu'aucun utilisateur n'y ait ouvert de session. Il suffit d'avoir la machine démarrée.

Toutes les opérations de LAUNCHER Office se feront ainsi en invisible. Si une application tente de rendre visible un document au travers de LAUNCHER Office en mode Service, le système génère une erreur.

LAUNCHER Office en mode service peut être installé sur tout PC utilisateur ou plus généralement sur un serveur Windows.

Installation / Désinstallation du service

1. Installation

Fermez toute exécution de LAUNCHER Office au préalable.

Pour installer LAUNCHER Office en service Windows, il faut lancer le programme **LNCsrv.exe** avec, en ligne de commande, l'option `/installservice`.

Utilisez une fenêtre Windows cmd et tapez la commande suivante :

"c:\program files\launcher<X>\lncsrv.exe" /installservice

Un message de confirmation doit apparaître : "Le service a été installé avec succès".

Le service sera installé, mais non démarré.

Il sera configuré en mode de démarrage **automatique**.

Ouvrez ensuite le gestionnaire de services pour démarrer le service une première fois.

Pour cela, allez dans le menu "Démarrer", "Panneau de Configuration", "Outils d'administration", "Services".

Faites un clic droit sur le nom "LAUNCHER/<X> for Windows NT" et cliquez sur "démarrer".

Le service démarre en quelques secondes et est prêt à l'utilisation.

2. Désinstallation

Pour désinstaller le service, il faut l'arrêter manuellement, puis utiliser la ligne de commande suivante :

"c:\program files\launcher<X>\lncsrv.exe" /uninstallservice



Contraintes du mode Service

Le mode service implique quelques contraintes.

Le service seulement

Il n'est pas possible d'utiliser LAUNCHER Office à la fois en service Windows et en mode Normal sur la même machine (sauf sur une machine TSE).

Si vous essayez de le faire vous aurez l'erreur suivante:

"L'application n'a pas réussi à s'initialiser correctement (0xc0000142). Cliquez sur OK pour arrêter l'application."

Pas d'affichage

L'application cliente qui pilote LAUNCHER Server ne doit pas faire d'affichage, sous contrainte de bloquer le service. Cependant un dispositif existe dans LAUNCHER Office pour limiter les risques de blocage et gérer les boîtes de dialogue bloquantes inévitables.

La fonction WORDSHOW, par exemple renverra une erreur.

Il peut être utile d'autoriser l'interaction avec le bureau en phase de développement (voir configuration du service par le gestionnaire de services de Windows).

Nom de machine

En mode LAUNCHER Office 'classique' on a pu prendre l'habitude d'utiliser l'identifiant '*DEV' pour la connexion par LNCOPEN.

Ici on devra désigner directement le **serveur** (par son adresse IP ou par son nom), puisque les programmes LAUNCHER Office ne seront plus exécutés depuis une émulation de terminal se trouvant sur la machine qui a le service en question.

Utilisateur 'SYSTEM'

Par défaut c'est l'utilisateur 'LocalSystem' ou 'SYSTEM' sous lequel tourne le service.

Cela peut être gênant lorsque l'on utilise la messagerie Lotus Notes ou MAPI, ou lorsque l'on veut utiliser une imprimante réseau.

Il peut être nécessaire de changer l'utilisateur dans la configuration du service, par le gestionnaire de services Windows.

Trace et configuration plus délicate

Le service n'a pas d'interface graphique. Il faut donc utiliser « LAUNCHER Administration » pour configurer le service et/ou consulter la trace. Pour la configuration, il est également possible de modifier directement le fichier Launcher.ini.

Le programme 'LAUNCHER Administration' vous permettra de configurer les options de LAUNCHER Office en service. Le nom du fichier .ini modifié est rappelé en premier onglet du programme. La trace est également consultable en temps réel par cet outil ou l'on peut opter pour une **trace de type « fichier »**.



Pour consulter la trace ou la configuration, utiliser l'onglet : « Configuration sur le réseau ».

Sélectionnez la ligne correspondant à la machine et au service.

Vérifiez dans le bas du premier onglet qu'il s'agit bien du service par le message :

"LAUNCHER Office Server version x.x.x.x – running as Service"

Ensuite on peut consulter la configuration ainsi que la trace.

Attention, certaines modifications des options de configuration réinitialisent la connexion avec l'AS/400.

L'arrêt, le démarrage ou le redémarrage du service ne peut se faire que par le gestionnaire de services de Windows.

Script de démarrage du service

Dans le cadre d'une utilisation de LAUNCHER Office en mode service Windows, il peut être utile de lancer des commandes à l'initialisation du service LAUNCHER Office.

Pour mettre en service un script de démarrage, il faut modifier le fichier launcher.ini comme suit:

```
[service]
```

```
StartupScript=c:\program files\launcher<X>\startup.cmd
```

(En supposant qu'il y ait un fichier startup.cmd dans le répertoire c:\program files\launcher<X>)

On peut insérer toute commande (lancement d'autres script, exécutable, mappage de lecteur réseau, ..) dans ce fichier. Ce script est traité ligne par ligne, et ne doit donc pas comporter de langage de script (if, goto, ...). Pour utiliser des scripts de ce type, les appeler depuis le script principal.

Si l'option est prise en compte, un rappel apparaît dans la trace de LAUNCHER Office.

Exemple :

```
Traitement du script de démarrage de service c:\program files\launcher400\startup.cmd)
```

Si une des lignes échoue, le texte de la ligne est rappelé dans la trace avec un message d'erreur.

Si le script bloque, le service ne démarrera pas. Il est donc impératif de tester le script "à part" avant de le mettre en place (par la commande SHELL de LAUNCHER Office par exemple).

NB : le script n'est exécuté que si Launcher Office est en mode service (option ignorée autrement).

Installation en multi instances

Vous pouvez installer plusieurs instances du serveur LAUNCHER Office sur un même serveur Windows.

Chaque instance doit avoir :

- Son propre service nommé, configuré et démarré.
- Son propre répertoire indépendant avec l'ensemble des fichiers programmes et DLL de LAUNCHER Office.
- Son propre port TCP/IP de communication. (LAUNCHER utilise par défaut le port 6078)
- Son propre fichier « launcher.ini » dans son répertoire propre.

Il faut installer chacun des services avec la commande suivante :



"<chemin de répertoire de instance>\Incsrv.exe" /installservice NomService /multiInstance /port numero

Exemple :

Soit deux répertoires LAUNCH_PROD et LAUNCH_DEV, contenant chacun une copie intégrale du répertoire initial « c:\Program Files\Launcher400 ».

```
"LAUNCHER_PROD\Incsrv.exe" /installservice Edit_Prod /MultiInstance /Port 6091
```

```
"LAUNCHER_DEV\Incsrv.exe" /installservice Edit_Dev /MultiInstance /Port 6092
```

Les deux commandes ci-dessus ont installé 2 services « Edit_Prod » et « Edit_Dev », respectivement sur les ports 6091 et 6092.

Pour adresser une des instances, les programmes clients doivent spécifier le numéro du port de l'instance.

Le port est donnée par le programme client :

- Avec l'adresse ou le nom du serveur, séparé par le caractère ":" :
NomDuServeur:Numéro
Exemple : Editique:6091

Exemples :

Connexion au service LAUNCHER_PROD :

```
LNCOPEN SVRADDR('Editique:6091')
```

Connexion au service Launcher par défaut :

```
LNCOPEN SVRADDR('Editique')
```

Boîtes de dialogue en mode "service Windows"

Les différentes boîtes de dialogue

Lorsqu'on est en mode service (sans interaction avec le bureau), des boîtes de dialogue (de Word ou Excel par exemple) peuvent apparaître pendant le traitement et bloquer les commandes LAUNCHER Office.

Cependant, LAUNCHER Office déclenche la fermeture de la boîte de dialogue automatiquement, ce qui évite le blocage.

LAUNCHER Office examine régulièrement les boîtes de dialogues présentes dans la session. Pour chaque boîte de dialogue "visible" (qui serait visible si on n'était pas en mode service Windows), des indications apparaissent dans la trace LAUNCHER Office.

Apparaît ensuite un texte décrivant l'action effectuée (ou rien si aucune action n'est effectuée).

LAUNCHER Office distingue deux types de boîtes de dialogue :

- **Les boîtes de dialogue** qui sont des boîtes d'information.
- **Les boîtes de message** qui sont des boîtes de question ou information.

Les boîtes de dialogue

Dans le cas d'une **boîte de dialogue**, LAUNCHER Office n'affiche que le **titre** de la boîte de dialogue ainsi que sa **classe**.

Exemple :

*Dialog info: **DialogTitle**=Enregistrer sous,
DialogClass=bosa_sdm_Microsoft Word 9.0, Style=94c80000,
StyleEx=501, Id=0.*

Une fenêtre popup avec le titre 'Enregistrer sous' est présente. Ceci peut bloquer LAUNCHER Office en mode 'service'.

LAUNCHER Office va fermer automatiquement cette fenêtre, et cela va probablement générer une erreur lors du traitement d'une commande LAUNCHER Office en cours.

La configuration de LAUNCHER Office permet de changer le comportement par défaut (ici la fermeture), en fonction du titre ou de la classe de la boîte de dialogue.

Les boîtes de message

Dans le cas d'une **boîte de message**, LAUNCHER Office affiche les mêmes informations que pour une boîte de dialogue "classique", mais avec en plus le **texte** du message ainsi que le texte des **boutons** normalement affichés à l'utilisateur.

Exemple :

*Dialog info: **DialogTitle**=Microsoft Word, **DialogClass**=#32770,
Style=94c801c5, StyleEx=10101, Id=0. **MsgBoxText**=Voulez-vous
enregistrer les modifications apportées à Document1 ?*

Button n.1 ID 6:'&Oui'

Button n.2 ID 7:'&Non'

Button n.3 ID 2:'Annuler'

Une fenêtre popup avec le titre 'Microsoft Word' est présente. Ceci peut bloquer LAUNCHER Office en mode 'service'. LAUNCHER Office va fermer automatiquement cette fenêtre, et cela va probablement générer une erreur lors du traitement d'une commande LAUNCHER Office en cours.

LAUNCHER Office a donc automatiquement fermé la boîte de message ce qui correspond à l'action "annuler", comme si un utilisateur avait appuyé sur la "croix" de la boîte de message.

La configuration de LAUNCHER Office permet de changer le comportement et de choisir un bouton à actionner automatiquement ou bien le bouton par défaut.

Configuration du comportement face aux boîtes de dialogue

Pour les **boîtes de dialogue** de type non "message", LAUNCHER Office ne peut pas récupérer le texte qui y est présent.

Les actions possibles sont "garder la boîte", "**fermer la boîte**", ou "valider la boîte".

"**Fermer la boîte**" est le comportement par défaut dans LAUNCHER Office (sauf pour les boîtes "impression en cours").

Seuls le **titre** et la **classe** peuvent donner une indication sur la boîte de dialogue.

Ainsi, deux types de paramétrage sont possibles :

- **sélection par titre**
- **sélection par nom de classe**

Remarque:

Les options qui suivent fonctionnent également pour les boîte de message.

Sélection par le titre

Pour mettre en place une action automatique selon le titre de la boîte de dialogue, utilisez les options suivantes dans le fichier launcher.ini :

```
[ServiceDialogs]
DialogTitle_<n>=<texte du titre>
DialogTitleAction_<n>=Close|Validate>
```

<n> est un incrément qui commence à 1

<texte du titre> est l'intitulé **exact** (casse respectée) du titre de la boîte de dialogue, correspondant à **DialogTitle=xxx** dans la trace LAUNCHER Office.

DialogTitleAction_<n> peut avoir comme valeur:

- "Keep" pour laisser la boîte de dialogue
- "Close" (par défaut) pour la fermer
- "Validate" pour valider

Exemple :

```
[ServiceDialogs]
DialogTitle_1=Enregistrer sous
DialogTitleAction_1=Validate
```

Ce qui aura pour effet de valider (et non fermer) toute boîte de dialogue ayant pour titre "enregistrer sous".

Après modification de launcher.ini (et **redémarrage** du service), la trace affichera pour le même cas de figure:

```
Dialog info: DialogTitle=Enregistrer sous, DialogClass=bosa_sdm_Microsoft Word 9.0, Style=94c80000, StyleEx=501, Id=0.
```

Le bouton par défaut de la boîte de dialogue 'Enregistrer sous' a été automatiquement sélectionné.

Sélection par la classe

Pour mettre en place une action automatique selon la classe de la boîte de dialogue, utilisez les options suivantes:

```
[ServiceDialogs]
DialogClass_<n>=<classe de la boîte>
DialogClassAction_<n>= <Keep|Close|Validate>
```

<n> est un incrément qui commence à 1 (numérotation séparée de DialogTitle_<n>).

<classe de la boîte> est l'intitulé **exact** de la classe de la boîte de dialogue.

DialogClassAction_<n> peut avoir comme valeur "Keep" pour laisser la boîte de dialogue, "Close" (par défaut) pour la fermer ou "Validate" pour valider.

Remarque:

Les numérotations de DialogClass_<n> et DialogTitle_<n> sont indépendantes (chacune commençant à 1).

Configuration du comportement face aux boîtes de message

Pour changer le comportement par défaut de LAUNCHER Office face à des boîtes de messages, les options possibles, dans Launcher.ini, sont les suivantes:

```
[ServiceDialogs]
MsgBoxText_<n>=<debut_du_texte>
MsgBoxActionNum_<n>=<numero_bouton>
```

<n> est un numéro qui commence à 1

<debut_du_texte> est le début du texte dans la boîte de dialogue (voir la valeur de "MsgBoxText=" dans la trace).

<numero_bouton> est le numéro du bouton à appuyer (commence à 1) ou 0 (par défaut) pour que le bouton par défaut soit sélectionné.

Exemple :

```
Dialog info: DialogTitle=Microsoft Word, DialogClass=#32770,
Style=94c801c5, StyleEx=10101, Id=0. MsgBoxText=Voulez-vous
enregistrer les modifications apportées à Document1 ?
Button n.1 ID 6:'&Oui'
```

Button n.2 ID 7: '&Non'
Button n.3 ID 2: 'Annuler'

Pour gérer ce cas, nous pouvons insérer:

```
[ServiceDialogs]
MsgBoxText_1=Voulez-vous enregistrer les modif
MsgBoxActionNum_1=2
```

Après modification de launcher.ini (et **redémarrage** du service), la trace affichera pour le même cas de figure:

```
Dialog info: DialogTitle=Microsoft Word, DialogClass=#32770,
Style=94c801c5, StyleEx=10101, Id=
MsgBoxText=Voulez-vous enregistrer les modifications apportées à
Document1 ?
Button n.1 ID 6: '&Oui'
Button n.2 ID 7: '&Non'
Button n.3 ID 2: 'Annuler'
```

Le bouton numéro 2 (libellé : '&Non') de la boîte de message 'Voulez-vous enregistrer les modifications apportées à Document1 ?' a été automatiquement sélectionné.

Pour faire plusieurs règles, incrémenter simplement <n>. Par exemple:

```
[ServiceDialogs]
MsgBoxText_1=Voulez-vous enregistrer les modif
MsgBoxActionNum_1=2
MsgBoxText_2=Texte deuxième boîte
MsgBoxActionNum_2=0
```

Combinaison de DialogClass_<n>, DialogTitle_<n> et MsgBoxText_<n>

Sachant qu'un nom de classe est assez souvent identique, on peut vouloir définir un comportement par défaut pour une classe donnée et affiner (gérer des "exceptions") par le titre.

Pour gérer cette possibilité LAUNCHER Office gère les options avec l'ordre de priorité suivant:

- 1) MsgBoxText_<n> : le plus prioritaire**
- 2) DialogTitle_<n>**
- 3) DialogClass_<n> : le moins prioritaire**

Par exemple, si on met dans le fichier launcher.ini:

```
[ServiceDialogs]
```



DialogClass_1=bosa_sdm_Microsoft Word 9.0

DialogClassAction_1=keep

DialogTitle_1=Enregistrer sous

DialogTitleAction_1=Validate

MsgBoxText_1=Voulez-vous enregistrer les modif

MsgBoxActionNum_1=2

Cela signifie que si une boîte de dialogue a pour classe "bosa_sdm_Microsoft Word 9.0", elle sera conservée à moins que le titre soit "Enregistrer sous". Auquel cas elle est validée.

A moins que ce soit une boîte de message dont le texte commence par " Voulez-vous enregistrer les modifications" auquel cas le bouton 2 est sélectionné.

Options générales concernant les boîtes de dialogue

Les options supplémentaires disponibles sont les suivantes :

[ServiceDialogs]

FilterMode=<0|**1**|2>

L'option "FilterMode" permet de d'activer ou de désactiver totalement la gestion automatique des boîtes de message. Les options possibles sont :

- 0** : Gestion des boîtes de dialogue totalement désactivée.
Toute boîte de dialogue sera conservée.
Les autres règles de la section [ServiceDialogs] sont ignorées.
- 1 (par défaut)** :Gestion "normale" des boîtes de dialogue.
Les autres options de la section [ServiceDialogs] seront utilisées.
Certaines règles internes sont utilisées en plus (par exemple pour ne pas fermer "impression en cours", mais fermer la boîte de dialogue "échec de publipostage").
- 2** : Toutes les boîtes de dialogues sont fermées par défaut.
Les autres règles de la section [ServiceDialogs] sont cependant utilisées.

Utiliser LAUNCHER Office

Démarrage du serveur LAUNCHER sur une machine Windows

Pour qu'une application cliente puisse communiquer avec un PC, il faut que le serveur LAUNCHER soit démarré sur le PC sous Windows.

Vous pouvez choisir de « **Créer un raccourci dans le groupe de démarrage** » pendant l'installation, pour que LAUNCHER soit lancé à chaque démarrage de Windows.

Sinon, lancez LAUNCHER Office sur le PC depuis le menu « Démarrer » :
« Démarrer » - « Programmes » - « LAUNCHER<X> » - « LAUNCHER<X> »
Il s'agit du mode "normal".

Sur un serveur Windows, vous pouvez [lancer le serveur LAUNCHER en mode "Service"](#).

Si Launcher Office est démarré en mode "normal", par défaut, le serveur Launcher lance aussi son administrateur.

Une icône apparaît sur la barre de tâches Windows.
En double cliquant sur cet icône, l'écran d'administration s'ouvre.

Le serveur LAUNCHER Office apparaît sous la forme du processus LNCSrv.exe, tandis que l'administrateur apparaît sous la forme du processus LNCAdm.Exe.

Utilisation du client AS/400

Sur l'**AS/400**, la **partie cliente** de LAUNCHER est représentée par un ensemble de programme, commandes et fichiers, tous réunis dans une seule librairie nommée "**LAUNCHER**" par défaut.

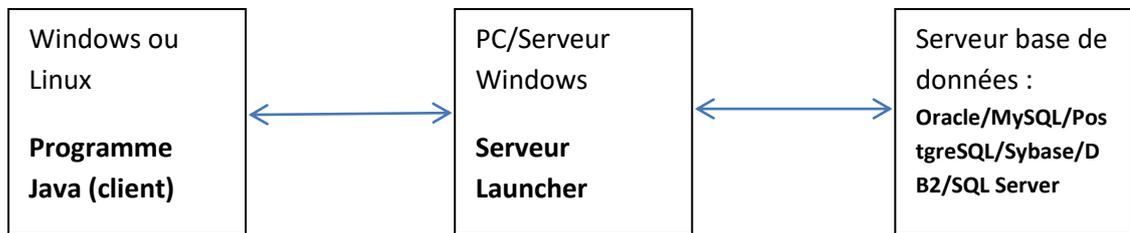
Cette librairie doit être en ligne pour les travaux qui utiliseront les programmes ou commandes de LAUNCHER :

ADDLIBLE LAUNCHER

Cette librairie peut être copiée ou renommée.

Il n'y a pas de configuration particulière à faire sur l'AS/400, à l'exception de celle imposée par [l'environnement Windows TSE ou CITRIX](#).

Utilisation du client Java



Le client Java utilisé dans l'application cliente permet une interaction avec le serveur Launcher.

Il permet d'utiliser:

- les commandes traditionnelles de Launcher avec **LNCCMD** ([voir liste des commandes](#))
- ou bien [les classes Java](#) fournies telles que **LNCTOXLS**, **LNCPRDOC** et **Datasource**.

Les différentes options pour LNCTOXLS et LNCPRDOC sont les mêmes que celles habituellement disponibles sous un environnement client OS400/System i.

La source de données en entrée des commandes LNCTOXLS et LNCPRDOC peut être soit :

- Un fichier au format CSV
- Soit issue des bases de données suivantes : Oracle, MySQL, PostgreSQL, Sybase, DB2 pour AS400 et Microsoft SQL Server.
- Soit issue d'un système SAP

Configuration de l'application Java côté client :

Les propriétés du projet Java, créé grâce à un IDE (environnement de développement intégré) du type Eclipse ou Netbeans..., doivent être modifiées afin la librairie suivante apparaisse dans le Java Build Path :

- Inciclient.jar

Il faut importer le JAR suivant :

C:\Program Files (x86)\LAUNCHER XPRESS\LAUNCHER_CLIENT\Inciclient.jar

Installation de SAP JCO sur le PC/serveur hébergeant le serveur Launcher:

Afin de pouvoir accéder au système SAP, il faut impérativement que le JAR sapjco.jar (version 2.1 de SAP JCo) et le JAR sapjco3.jar (version 3.0 de SAP JCo) soient présents dans le répertoire d'installation du serveur Launcher.

Par exemple :

C:\Program Files (x86)\LAUNCHERi\LAUNCHER_CONNECTORS\Inciconnector_lib.

De plus il faut également que les DLLs associées au connecteur SAP JCo 2.1 (sapjcorfc.dll et librfc32.dll) ou SAP JCo 3.0 (sapjco3.dll) soient présentes dans le répertoire suivant :

C:\Program Files (x86)\LAUNCHERi\LAUNCHER_CONNECTORS\Inciconnector_lib

Installation des JARs JDBC sur le PC/serveur hébergeant le serveur Launcher:
Afin de pouvoir accéder aux bases de données Oracle, MySQL, PostgreSQL, Sybase, DB2 pour AS400 ou Microsoft SQL Server, il faut impérativement que les JARs JDBC des bases de données soient présents dans le répertoire d'installation du serveur Launcher.

Par exemple :

C:\Program Files (x86)\LAUNCHERi\LAUNCHER_CONNECTORS\Inciconnector_lib.

Liste des JARs JDBC officiels à utiliser **obligatoirement** (aucune garantie de fonctionnement pour tout autre JAR utilisé):

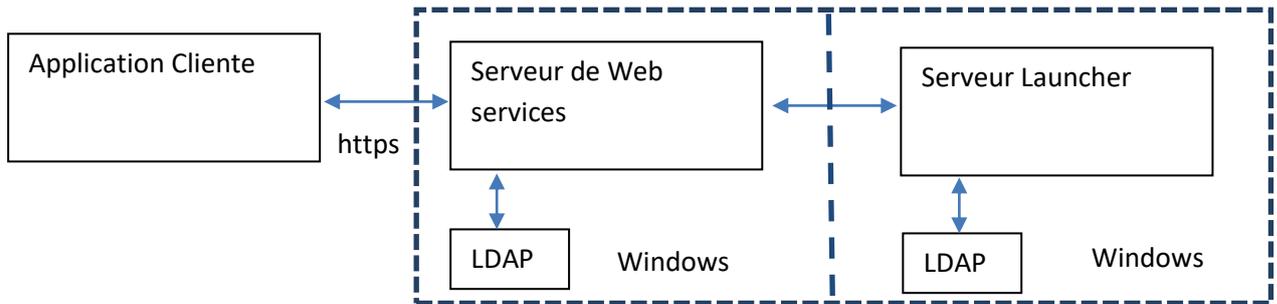
- Pour les bases de données Oracle, il faut utiliser le driver Thin fourni par Oracle. Par exemple pour Oracle Database 10g Release 2 , le driver Thin JDBC est nommé **ojdbc14.jar** , et peut être téléchargé en utilisant le lien suivant :
<http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-10201-088211.html>
- Pour les bases de données MySQL, le driver JDBC **mysql-connector-java-5.1.21-bin.jar** doit être utilisé :
<http://www.mysql.com/downloads/connector/j/>
- Pour les bases de données PostgreSQL (≥ version 7.2), le driver JDBC **postgresql-9.2-1000.jdbc4.jar** peut être téléchargé en suivant ce lien :
<http://jdbc.postgresql.org/download.html>
- Pour les bases de données Sybase (tous les produits Sybase utilisant JConnect-7_0: Adaptive Server Enterprise, SQL Anywhere, Sybase IQ, et Replication Server), le driver **jconn4.jar** doit être utilisé :
<http://www.sybase.fr/products/allproductsa-z/softwaredeveloperkit/jconnect>
- Pour les bases de données Microsoft SQL Server (SQL Server2012, SQL Server 2008 R2, SQL Server 2008, SQL Server 2005, et SQL Azure), le driver **sqljdbc4.jar** doit être utilisé :
<http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774>
- Pour les bases de données IBM DB2 pour iSeries, il faut utiliser le driver **jt400.jar** fourni avec IBM System i Access for Windows. Après avoir installé IBM Access sur votre PC, vous pourrez trouver le driver JDBC par exemple ici :
C:\Program Files (x86)\IBM\Client Access\jt400\lib\jt400.jar

Après avoir installé les JARs JDBC, il faut absolument redémarrer le serveur Launcher.



Utilisation des webservices

Avec la fonctionnalité Web services de Launcher, la partie cliente peut être une application exécutée sur n'importe quelle plateforme. Cette application appelle des webservices pour passer des commandes au serveur Launcher. **Un serveur de webservices fait la passerelle entre l'application cliente et le serveur Launcher :**



Les webservices peuvent être utilisés avec n'importe quel langage, sur n'importe quelle plateforme (Windows, Linux).

Configuration

Une authentification LDAP (Active Directory) est nécessaire entre l'application cliente et le serveur de Webservices. Elle se fait par le biais de la méthode [open](#).

Dans le fichier "**C:\Program Files (x86)\LAUNCHER XPRESS\REST_GATEWAY\configs\authorizedUsers.xml**", il faut spécifier le nom du domaine sur lequel tournent le serveur de Webservices et le serveur LDAP (Active Directory) :

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?><authentication>
  <!-- authentication_mode : "ldap" for Active Directory -->
  <authentication_mode>ldap</authentication_mode>

  <ldap>
    <domain>aura.fr</domain>
  </ldap>
</authentication>
```

L'authentification LDAP sera utilisée :

- pour les communications entre l'application cliente et le serveur de webservices

[Mise en place de l'authentification LDAP pour le serveur Launcher:](#)

Dans le cas où le serveur Launcher n'est pas sur la même machine Windows et éventuellement pas sur le même réseau, il y a la possibilité de mettre en place une authentification LDAP (Active Directory) lors de la connexion du Serveur de Webservices au serveur Launcher.

Dans le fichier **launcher.ini** (*C:\Program Files (x86)\LAUNCHER XPRESS\launcher.ini*), il faut mettre « UseLdap=Yes » et le nom du domaine est spécifié :

```
[LDAP]
UseLdap=Yes
Hostname=aura.local
```

L'authentification LDAP sera utilisée :

- pour établir la communication entre le serveur de webservices et le serveur Launcher. Cette authentification s'établit avec la commande **LDAPAUTH** et doit impérativement suivre la commande **LNCOPEN**. Elle ne sera plus prise en compte dès la fin de la communication provoquée par la commande **LNCCLOSE**.

Utilisez une fenêtre terminal Windows "cmd" en tant qu'administrateur et exécutez les commandes suivantes:

```
cd C:\Program Files (x86)\LAUNCHER XPRESS
notepad launcher.ini
```

Redémarrez ensuite le serveur Launcher.

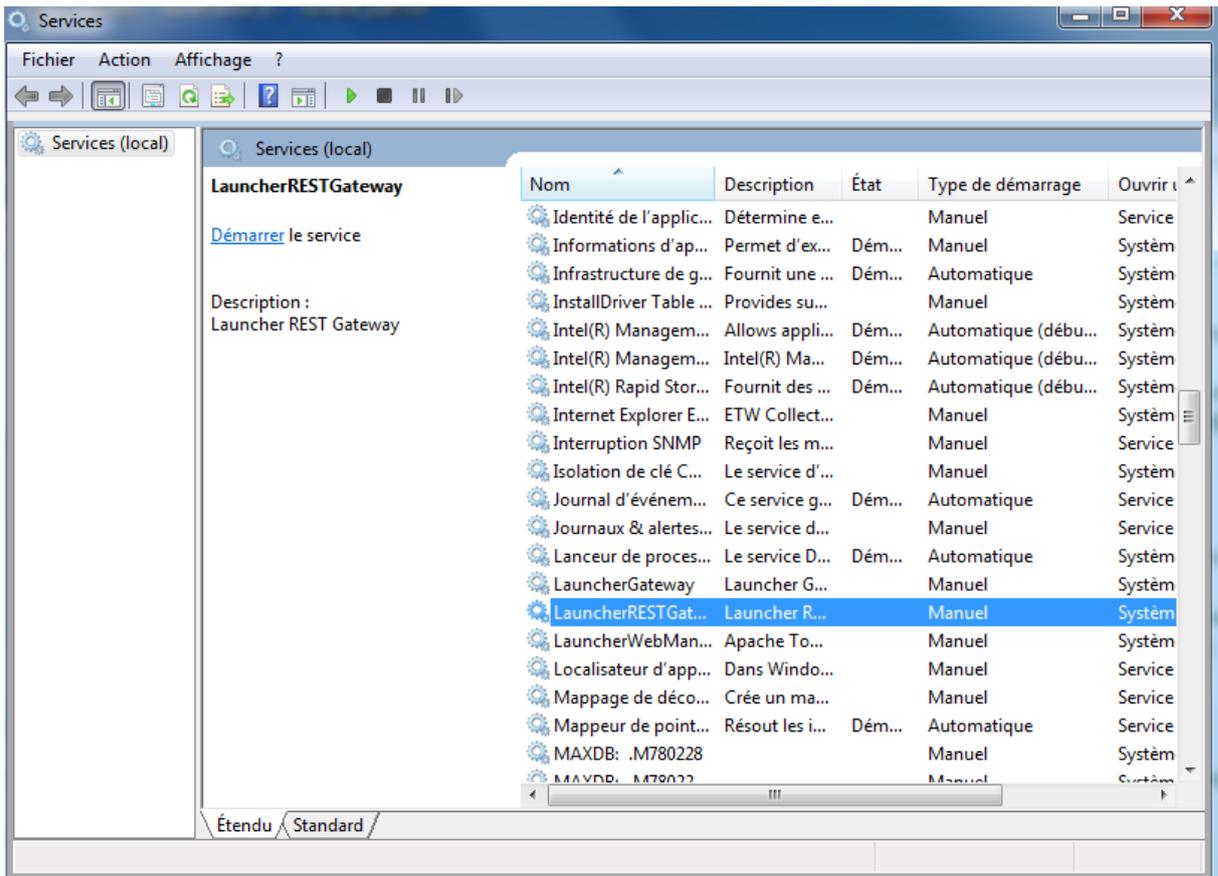
Pour ne pas mettre en place d'authentification LDAP, il suffit de mettre « UseLdap=No » :

```
[LDAP]
UseLdap=No
```

Démarrer le serveur de webservices : Launcher REST Gateway

« **Launcher REST Gateway** » a été installé en tant que **service Windows**.

Allez dans : Panneau de configuration\Tous les Panneaux de configuration\Outils d'administration\Services :



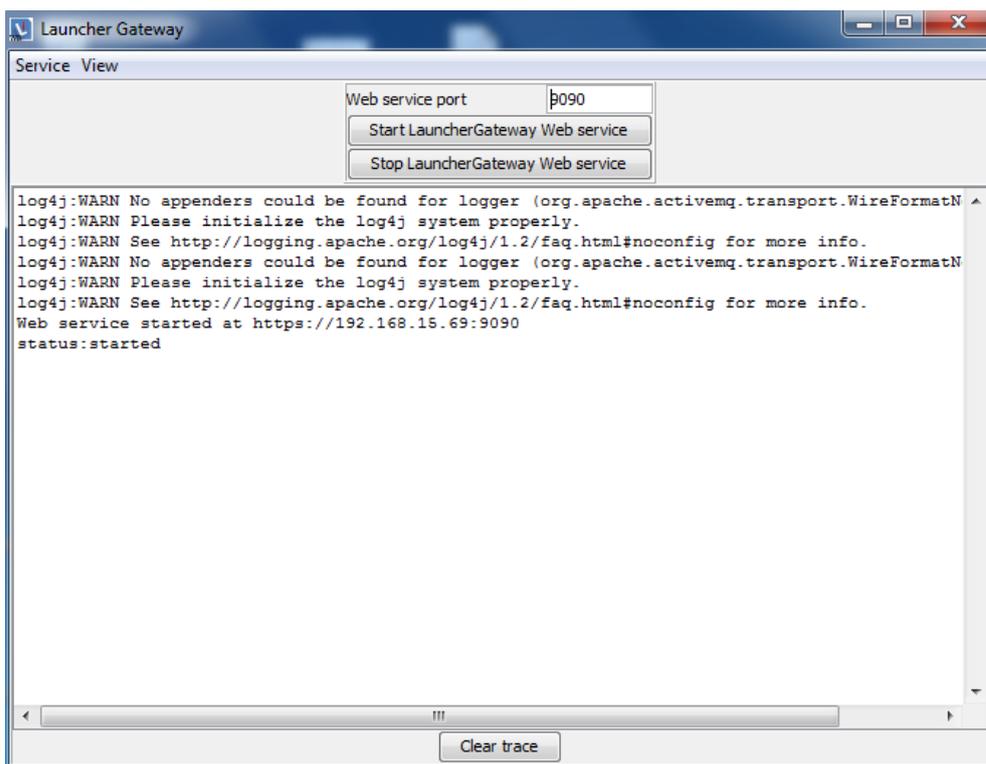
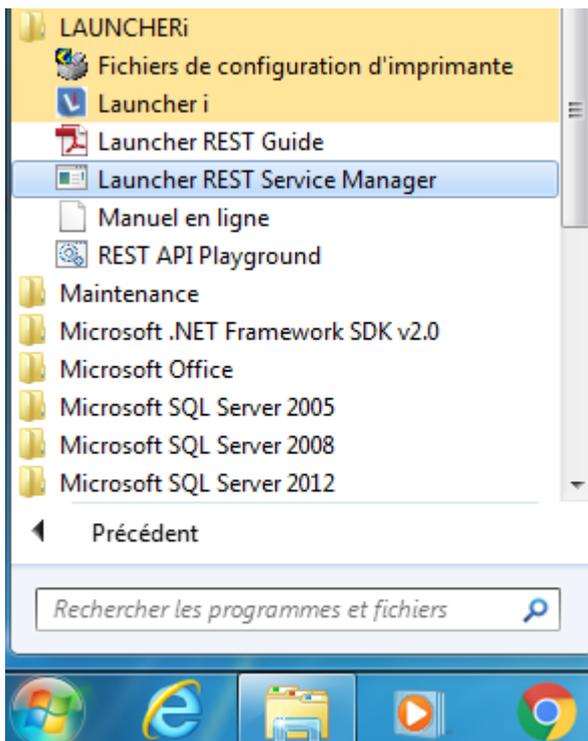
Démarrez le serveur de webservices « Launcher REST Gateway ».

Remarque :

Le serveur de webservices doit être arrêté à partir de l'application Launcher REST Service Manager.

Launcher REST Service Manager

A partir du groupe de démarrage LAUNCHER XPRESS, lancez « **Launcher REST Service Manager** » :



On voit que la Gateway Launcher REST est bien démarrée.

Le serveur Launcher peut donc recevoir des commandes à exécuter de l'application cliente, par le biais du serveur de webservices.

Cette console permet :

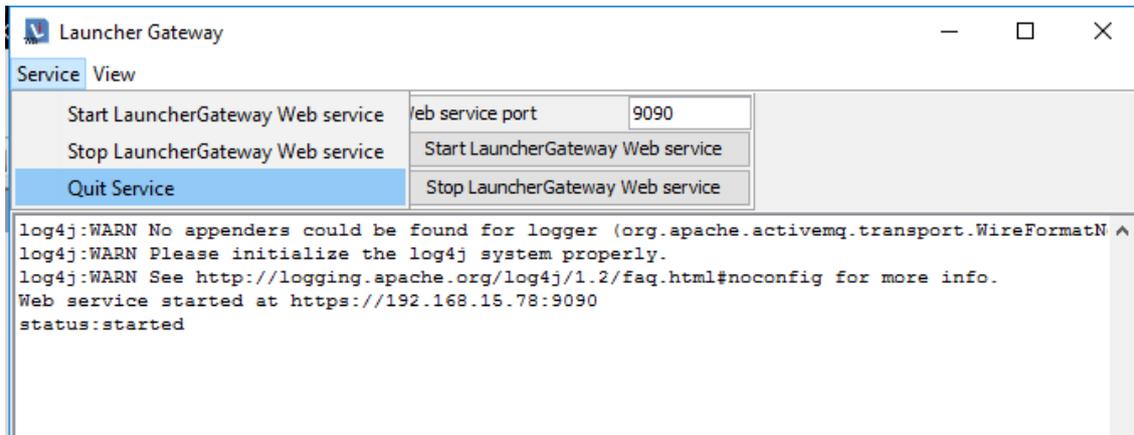
- d'avoir une trace de l'activité du serveur de webservices.

- d'arrêter le serveur de webservices.

Arrêter le serveur de webservices : Launcher REST Gateway

Le seul moyen d'arrêter correctement le serveur de webservices, est d'utiliser l'application Launcher REST Service Manager à partir du menu Démarrer.

Il faut cliquer sur "Quit Service" dans le menu Service :



Launcher Administration

LAUNCHER Office est constitué d'un programme qui doit être lancé sur chaque PC destiné à recevoir des requêtes clientes.
Le PC héberge donc la partie serveur de LAUNCHER Office.

Ce programme est subdivisé en deux processus :

- l'exécutable serveur (LNCsrv.exe), qui reçoit les requêtes clientes
- l'exécutable d'administration (LNCadm.exe), qui permet d'agir sur la configuration du serveur, et d'avoir une vue sur son activité avec la trace.

Exécutez le raccourci "LAUNCHER<X>" situé dans l'entrée LAUNCHER<X> du menu Démarrer. Ce raccourci lance LNCsrv.exe qui lance lui-même LNCadm.exe.

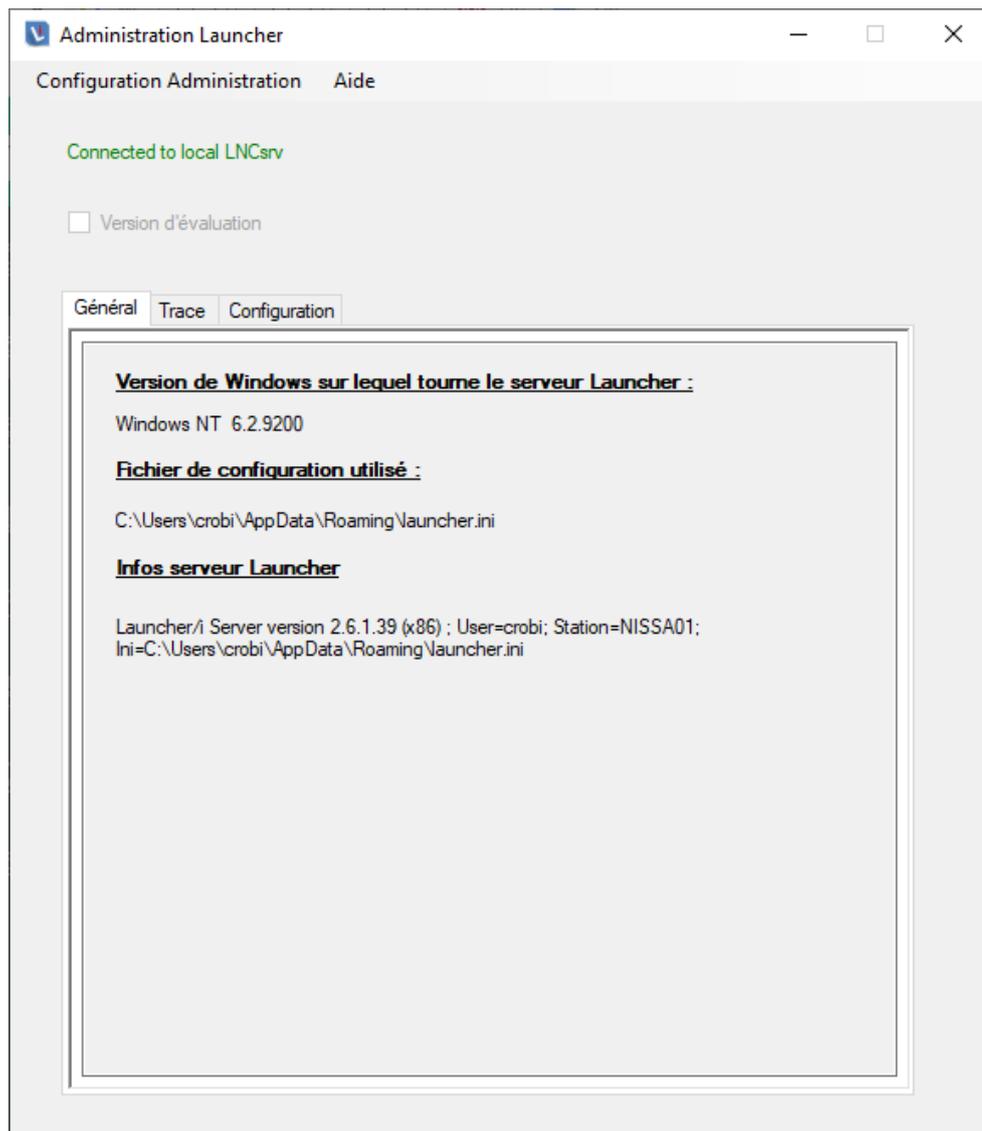
Si LNCsrv.exe est déjà lancé sur le poste, il n'est pas relancé, et la fenêtre d'administration (LNCadm.exe) apparaît, en mode de configuration locale.

Ces programmes s'exécutent en tâche de fond.

Le programme LNCsrv.exe est en attente de connexion du client.

Pour que **LAUNCHER Office** s'exécute systématiquement à l'ouverture de Windows, copier le raccourci "Launcher<X>" se trouvant dans l'entrée Launcher<X> du menu démarrer vers l'entrée "Démarrage".

La fenêtre d'administration de **LAUNCHER Office** est découpée en trois onglets : "**Général**", "**Trace**", et "**Configuration**".



L'onglet Trace

Cet onglet contient toutes les instructions envoyées par le client : les demandes de connexions, les commandes et leurs paramètres et le cas échéant les descriptions d'erreurs.

Un message spécifie que le serveur est prêt à recevoir une demande de connexion du serveur ("*Attente de connexion*").

Lorsque la connexion est établie entre les deux parties ("*Connexion acceptée*"), l'échange de messages peut avoir lieu ("*Début du traitement des messages*").

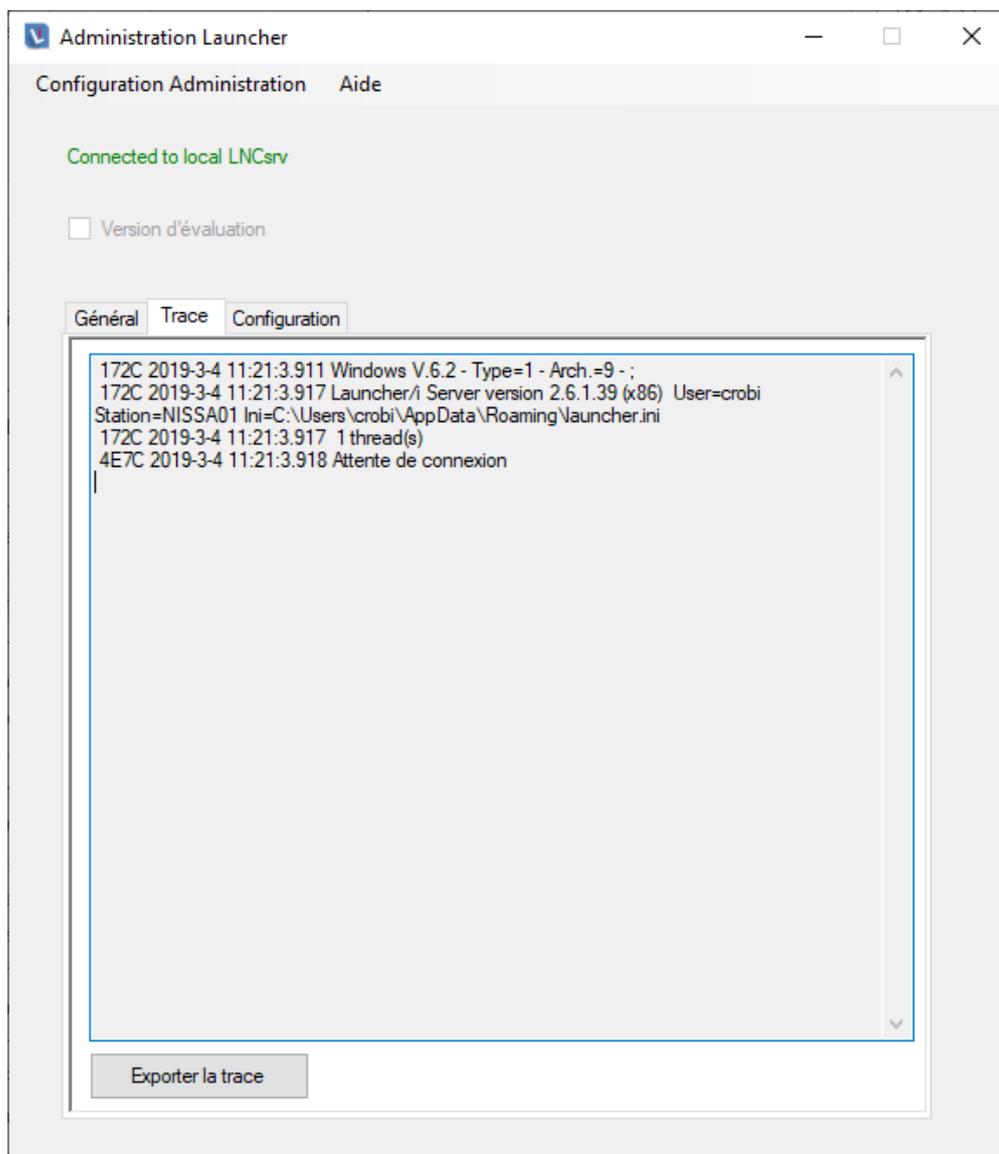
Pour toutes les commandes reçues, le PC va afficher le nom de la commande, l'option, les deux paramètres (Parm1 et Parm2) ainsi que la taille de chaque élément. Exemple : Parm1(33) et Parm2(74). Enfin, les éventuelles erreurs sont également affichées.

Une fois la commande END envoyée par le client, le PC ferme la connexion ("*Fin de la transmission*"), puis repasse en attente de connexion ("*Attente d'une connexion*").

Il n'y a pas de limite sur le volume de trace affichée (mémoire de la machine uniquement). Par contre seules les 500 dernières lignes apparaissent.

Il peut dès lors être judicieux de spécifier un fichier PC pour enregistrer la trace.

L'onglet "Configuration" permet d'afficher l'heure et/ou le numéro de thread sur chaque ligne de trace, ou bien de doubler la trace dans un fichier sur le PC, comme évoqué ci-dessus.



L'onglet Configuration

Cet onglet permet de paramétrer le fonctionnement global de l'application.

Les différentes options présentes vous permettent de faire les réglages suivants :

- **Démarrer Word invisible** : pour travailler sans gêner une autre utilisation, cette instance reste cachée jusqu'à un éventuel appel à la fonction [WORDSHOW](#).
- **Démarrer Excel invisible** : pour travailler sans gêner une autre utilisation, cette instance reste cachée jusqu'à un éventuel appel à la fonction [EXCELSHOW](#).
- **Système de messagerie** : va vous permettre d'utiliser les commandes de messagerie pour envoyer des courriers électroniques (voir le chapitre "[Programmation avancée/Mails](#)" pour plus d'informations).
- **Fermer la connexion en cas d'erreur** : lorsque cette option est cochée, la connexion est automatiquement fermée si une erreur quelconque survient.
- **Configuration de la trace** : permet de désactiver totalement le système de trace, de choisir une trace vers un fichier PC, ou bien de rajouter le numéro de thread et/ou la date et l'heure de chaque opération du serveur.
- **Mode Terminal services** : permet de basculer en mode "Terminal services". Lorsque cette option est active on doit choisir l'adresse IP de l'AS/400 ainsi qu'un nom pour le poste. Voir "[Configuration de LAUNCHER Office pour Windows Terminal Server](#)".
- **Démarrer LAUNCHER Office en mode invisible** : si cette option est cochée, le lancement de LAUNCHER Office par l'icône classique ne lancera pas l'administration de LAUNCHER Office. Dans ce mode, il faut lancer manuellement l'administration de LAUNCHER Office pour pouvoir changer la configuration ou accéder à la trace mémoire.

Administration Launcher

Configuration Administration Aide

Connected to local LNCsrv

Version d'évaluation

Général Trace Configuration

Trace

Trace active

Afficher le n° de Thread

Afficher la date/heure

Trace dans un fichier sur le PC

Microsoft Office

Démarrer Excel invisible

Démarrer Word invisible

Transfert des données en Unicode par défaut

Email

SMTP v

Mode TSE

Désactivé

Utiliser nom utilisateur

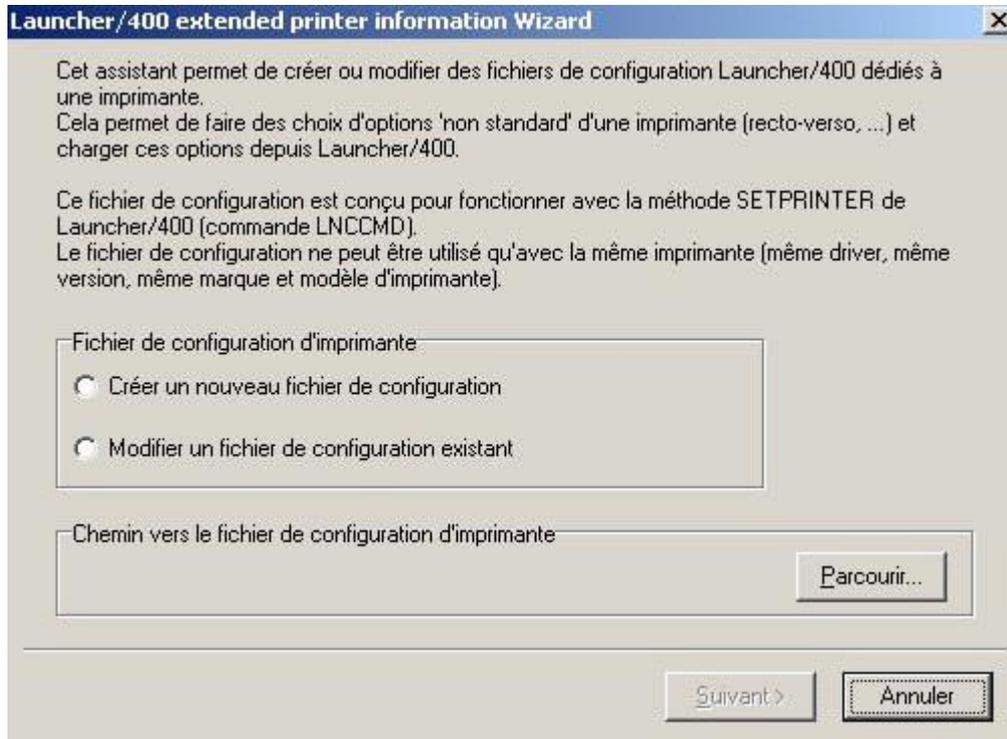
Utiliser nom station

Utiliser nom fixe

Appliquer
Annuler

Fichiers de configuration d'imprimante

Cet assistant permet de créer ou modifier des fichiers de configuration dédiés à une imprimante.



Voir :

[Commande SETPRINTER](#)

[Commande RSTPRINTER](#)

Administration à distance

Lors de l'installation vous pouvez sélectionner "Outils d'administration de LAUNCHER Office", à condition d'avoir les **droits d'administration** au moment de l'installation.

Pour lancer l'administration à distance, utiliser le menu 'Démarrer' 'Launcher<X>' "Launcher<X> – Administration".

L'interface utilisateur ressemble à l'interface de l'outil d'administration lancé avec l'icône "Launcher<X>", mais comporte des onglet supplémentaires :

- **"Configuration locale"** : permet d'administrer l'instance de LAUNCHER Office locale, invisible ou non.
- **"Configuration sur le réseau"** : permet d'administrer à distance n'importe quel LAUNCHER Office du réseau local.

L'administration à distance implique plusieurs restrictions :

- l'utilisateur doit avoir les droits d'administration réseau pour voir apparaître les onglets supplémentaires.
- l'instance de LAUNCHER Office à administrer doit être en cours d'exécution et iconisée pour pouvoir être administrée à distance.
- l'instance de LAUNCHER Office à administrer ne doit pas déjà être en cours d'administration

Lorsque l'onglet "**Configuration sur le réseau**" est sélectionné, la liste des instances de LAUNCHER Office qui peuvent être administrées apparaît (suivant les critères énoncés ci-dessus). Il suffit alors de cliquer sur la machine pour accéder à sa configuration.

Cette liste ne contient que la machine du masque réseau local. Pour administrer d'autres machines, il faut modifier le fichier launcher.ini de la machine à administrer (voir chapitre '[Fichier de configuration](#)').

L'onglet général montre la version du système, la version de LAUNCHER Office, et l'emplacement du fichier launcher.ini.

L'onglet général montre également le mode de fonctionnement courant : connecté ou non connecté. Si l'on est en mode non connecté, cela signifie que l'on a pas trouvé d'instance de LAUNCHER Office suivant l'état actuel (par exemple pas de LAUNCHER Office en cours d'exécution sur la machine locale). Dans ce cas la version de l'administrateur apparaît.

On peut lire et modifier la configuration à distance. Attention, la modification de certaines options implique le redémarrage **automatique** de LAUNCHER Office, ce qui signifie **coupure de la connexion** vers le client. Cela peut donc avoir des conséquences si LAUNCHER Office est en cours d'utilisation.

Les propriétés qui provoquent ce redémarrage automatique au moment de l'application des changements sont les suivantes :

- mode de connexion Terminal Services
- nom local Terminal Services (en mode Terminal Services)



- nom ou adresse IP de l'AS/400 (en mode Terminal Services)
- type de messagerie
- mot de passe Lotus Notes
- nom de profil MAPI

Il est également possible d'administrer une instance de LAUNCHER Office en Service NT, sur le poste local ou sur un poste du réseau local.

Si vous êtes sur un serveur Terminal Services, un onglet supplémentaire apparaît : "**configuration des utilisateurs**" qui permet de modifier les fichiers launcher.ini locaux de chaque utilisateur qui s'est déjà connecté à un moment donné sur la machine (ici LAUNCHER Office n'a pas besoin de tourner).

Si on est en mode non connecté, (par exemple si l'instance locale de LAUNCHER Office n'est pas trouvée), on peut néanmoins modifier la configuration, qui va dans ce cas écrire directement dans le fichier launcher.ini. C'est le mode de fonctionnement lorsque l'on est sur l'onglet "configuration des utilisateurs".

Fichier de configuration

Toute la configuration modifiable par l'administrateur de LAUNCHER Office réside dans le fichier **launcher.ini**

Ce fichier **launcher.ini** est utilisé par LAUNCHER Office pour connaître la configuration.

Le programme d'administration de LAUNCHER Office indique l'emplacement exact du fichier **launcher.ini** actuellement utilisé, sur l'onglet principal.

(voir : [Administration de Launcher Office](#))

Au premier lancement de Launcher (à partir du menu Démarrer/LAUNCHER<X>/Launcher<X>), le fichier **launcher.ini** utilisé est :

C:\Users\<<user>\AppData\Roaming\launcher.ini.

Il sera créé à cet endroit si des modifications dans la configuration de Launcher sont opérées, et qu'aucun fichier **launcher.ini** n'existe.

Sinon par ordre de priorité décroissante, **launcher.ini** est recherché dans les emplacements suivants:

- C:\Program Files (x86)\LAUNCHER<x> (ou C:\Program Files\LAUNCHER<x>)
- C:\Windows
- C:\Users\<<user>\AppData\Roaming

Si le profil utilisé pour ouvrir la session Windows n'a pas les droits pour modifier le fichier **launcher.ini**, alors les cases de la configuration sont grisées : aucune modification possible. Il faut lancer Launcher "en tant qu'administrateur" pour modifier la configuration.

Si Launcher est démarré en tant que **service Windows**, **launcher.ini** est recherché dans les emplacements suivants par ordre de priorité décroissante:

- C:\Program Files (x86)\LAUNCHER<x> (ou C:\Program Files\LAUNCHER<x>)
- C:\Windows

Il sera créé dans "C:\Windows" si des modifications dans la configuration de Launcher sont opérées, et qu'aucun fichier **launcher.ini** n'existe.

Si la configuration d'un serveur Launcher est modifiée **à distance** avec "Launcher<x> - Administration" (à partir du menu Démarrer/LAUNCHER<x>),

launcher.ini est recherché dans les emplacements suivants:

- C:\Program Files (x86)\LAUNCHER<x> (ou C:\Program Files\LAUNCHER<x>)
- C:\Windows
- C:\Users\<<user>\AppData\Roaming

Depuis la version 2.6.1.22, il est possible de passer en paramètre le chemin du fichier launcher.ini.

Exemple : Incsrv.exe /ini "C:\temp\launcher.ini"



Le contenu de **launcher.ini** est modifié par le programme d'Administration de LAUNCHER Office, mais certaines options ne sont disponibles qu'en modifiant directement le fichier.

Ces options sont les suivantes :

- **Section [CONFIG]**
 - **MaxThreads=n**

Définit le nombre de connexions simultanées maximum (voir "[mode multi connexions](#)"). Valeur par défaut : 1
- **Section [NETWORK]**
 - **PortNumber=n**

Numéro de port TCP/IP de LAUNCHER Office. Valeur par défaut : 6078
 - **LNCD_PortNumber=n**

Numéro de port TCP/IP de LAUNCHERD. Valeur par défaut : 5681
 - **Admin_PortNumber=n**

Numéro de port TCP/IP (UDP) utilisé pour l'administration à distance de LAUNCHER Office. Valeur par défaut : 6079
 - **Admin_PortNumber2=n**

Numéro de port TCP/IP (TCP) utilisé pour l'administration à distance de LAUNCHER Office. Valeur par défaut : 6080
 - **Admin_PortNumber2End**

Numéro de port TCP/IP (TCP) maximum utilisé pour l'administration à distance de LAUNCHER Office. Valeur par défaut : 6085. Est utilisé en cas d'administration multiple sur serveur TSE. On peut augmenter la valeur pour admettre plus d'administrations simultanées sur le serveur.
 - **RetriesDelay**

Délai en secondes entre deux tentatives de connexions à LAUNCHERD. Valeur par défaut : 30. N'est utilisé qu'en mode TSE.
 - **AdminFromIPAllowxx=ipaddress**

Indique une ou des adresses IP depuis la(les)quel(les) il est possible n'administrer le poste LAUNCHER Office. Ceci est utile pour pouvoir administrer une machine qui ne se trouve pas sur le même masque réseau. Vous pouvez indiquer plusieurs adresses ou bien indiquer un réseau entier; dans ce cas, incrémenter le préfixe derrière l'option.

Par exemple:
[NetWork]
AdminFromIpAllow1=192.206.160.5
AdminFromIpAllow2=192.206.161.255

Dans cet exemple, la machine '192.206.160.5' et toutes les machine commençant par '192.206.161.' sont autorisées à utiliser l'administration à distance vers cette machine.

NB: quoi qu'il arrive, toutes les machines de la branche 'locale' du réseau ont l'autorisation de pratiquer l'administration à distance.



Utilisez également l'option AdminToIPxx sur la machine d'administration.

- **AdminToIPxx=ipaddress**

Indique l'adresse IP d'une ou plusieurs machines que l'on veut administrer à distance.

Par défaut, l'administrateur de LAUNCHER Office peut accéder aux instances de LAUNCHER Office se trouvant sur le même masque réseau que la machine locale. Cette option permet d'administrer d'autres machines, à condition que celles-ci **acceptent la connexion depuis la machine où l'administrateur de LAUNCHER Office doit être lancé** (voir option 'AdminFromIPAllowxx').

Par exemple :

[Network]

AdminToIP1=192.206.162.4

AdminToIP2=192.206.162.5

Dans cet exemple, on veut pouvoir administrer les machines 192.206.162.4 et 192.206.162.5.

Remarque: ici on ne peut pas mettre d'adresse générique (comme par exemple 192.206.162.255)

Windows Terminal Server et AS400

Cette configuration n'impose pas nécessairement Windows Terminal Server ou Citrix.

Quand les PC des utilisateurs n'ont pas une adresse IP pour chacun d'eux, ou quand l'émulation 5250 passe au travers d'un frontal qui masque l'adresse IP du PC, la configuration en "Mode TSE" de LAUNCHER Office peut être utilisée.

En configuration standard (TSE désactivé), le serveur Launcher sur le PC est en attente d'un appel arrivant de l'AS/400.

Le programme AS/400 doit connaître l'adresse IP du PC, ou son nom DNS, pour le contacter par LNCOPEN.

Par la valeur spéciale "*DEV", LAUNCHER récupère l'adresse IP du PC dans la description de l'écran.

Si l'adresse IP du PC ne peut pas être connue facilement du programme AS/400, ou, si plusieurs utilisateurs ont la même adresse (dans le cas de Citrix ou TSE), alors, on inverse le sens de conversation en configurant LAUNCHER Office pour le mode TSE.

Le mode TSE se configure par l'écran de configuration du serveur LAUNCHER sur le PC, onglet « Configuration ».

Note : Les deux modes de communication peuvent cohabiter autour d'un même AS/400.

Particularités d'installation AS/400

- Sur AS/400, lancez le démon LAUNCHERD :

ADDLIBLE LAUNCHER
STRLNCD MSGLVL(1) <F4>

Lancez ce programme sous un profil ayant des droits sur les autres, ou modifiez le programme LAUNCHERD de LAUNCHER pour qu'il tourne avec les droits de QSECOFR.

CHGPGM LAUNCHER/LAUNCHERD USRPRF(*OWNER)

Un programme "démon" **LAUNCHERD** sera lancé.

Si vous laissez la valeur par défaut pour JOBQ, LAUNCHERD sera démarré dans le sous système QSYSWRK.

La commande **DSPLNCD** permet de visualiser les postes connectés.

Particularités sur le serveur TSE

L'utilisation de LAUNCHER Office sur un serveur "Terminal Services" valable pour tous les utilisateurs, implique une instance de LAUNCHER Office qui va fonctionner pour chaque utilisateur.

Ces instances peuvent venir en plus d'une autre instance de LAUNCHER Office sous forme de service NT.

Afin de gérer ces multiples instances de LAUNCHER Office, on ne peut plus identifier un poste LAUNCHER Office par sa seule adresse IP.

Chaque utilisateur TSE de LAUNCHER Office doit avoir un **nom unique**, et désigner l'**adresse IP l'AS/400**.

Donc, dans l'écran de configuration, il faut sélectionner un des modes "Terminal Services" (onglet Configuration).

Les options pour le choix du nom sont :

- **Utiliser le nom de l'utilisateur** : le nom WINDOWS de l'utilisateur sera utilisé comme nom.
- **Utiliser le nom de la station** : le nom du PC connecté à TSE (poste client) sera utilisé comme nom.
- **Utiliser un nom fixe** : permet de choisir un nom quelconque (à vous de vous assurer de l'unicité de ce nom sur le serveur TSE).

Votre application devra ensuite gérer les noms, en utilisant par exemple le nom de l'utilisateur comme nom local unique, à condition qu'un utilisateur ne se connecte que depuis une seule station.

Quand le mode TSE est configuré sur l'AS/400 (LAUNCHERD est actif), lors d'un appel à LNCOPEN, LAUNCHER procède à la recherche du PC selon l'ordre suivant :

- Recherche du PC portant le nom "TSE" demandé.
- Recherche du PC ayant le nom "DNS" demandé.

Si vous configurez l'émulateur 5250 de la même manière (exemple de nom de station), vous pouvez utiliser le nom * DEV dans la commande LNCOPEN. Sinon, utilisez le nom correspondant.

Quand **"*DEV"** est utilisé, alors que LAUNCHERD est actif, Launcher recherche le PC comme suit :

- Recherche du PC ayant pour nom "TSE" le nom du travail en cours (Voir ci après : Filtrage du nom du travail en cours).
- Recherche du PC par l'adresse IP associée au device écran.

Filtrage du nom du travail en cours :

Si vous configurez l'émulateur 5250 de Client Access pour que le nom de terminal soit issu du nom du poste, vous pourrez utiliser la valeur spéciale **"*DEV"** en mode TSE ou non TSE.

Changez le contenu de la « data area » LNCNET de la librairie LAUNCHER, position 1 à 100.

```
CHGD TAARA DTAARA(LNCNET (1 100)  
VALUE('AUTO;STNNAME= <filtre>');
```

Par la valeur de <filtre> vous allez indiquer comment Client Access a ajouté des préfixes ou suffixes au nom de la station, pour construire le nom du terminal.

<filtre> peut prendre les valeurs suivantes :

- **+&COMPN** Aucun préfixe ou suffixe n'est ajouté.
- **%+&COMPN** Un caractère préfixe ajouté pour identifier une émulation écran ou imprimante.
- **+&COMPN*** Un suffixe ajouté pour éviter noms en double sur le poste.
- **+&COMPN=** Un suffixe ajouté pour éviter noms en double avec d'autres postes.
- **+&COMPN*=** Deux suffixes ont été ajoutés.
- **%+&COMPN*=** Un préfixe et les deux suffixes ont été ajoutés.

Note : Ces valeurs sont celles utilisées par Client access dans son fichier de définition d'écran (Fichier .WS).

Exemple :

La station Windows se nomme **PAUL**.

Le serveur Launcher sur le PC est configuré en mode TSE, et il utilise le nom de la station comme nom Launcher.

Client access est configuré pour ajouter le préfixe et les deux suffixes. Le nom du travail en cours est : **SPAULA1**

La data area doit contenir :

AUTO;STNNAME=%+&COMPN*= ;

Note :

Si le nom est générique (par le nom de l'utilisateur ou de la station), il est possible de configurer LAUNCHER Office de manière unique pour TOUS les utilisateurs. Pour cela il suffit de configurer sur un poste, puis de copier le fichier launcher.ini de l'utilisateur dans le répertoire de LAUNCHER Office.

Voir chapitre "Launcher.ini".

Si vous rencontrez des problèmes, affichez la trace PC, et consultez le fichier de message sur AS/400 :

DSPMSG LAUNCHER/LNCSMSGQ

Mode Multi Connexions

LAUNCHER Office a la possibilité de recevoir des requêtes **simultanées**.

Ces requêtes pourront venir de plusieurs programmes clients fonctionnant en même temps.

Vous devez choisir le nombre maximum de connexions simultanées par l'option suivante dans **launcher.ini**:

[Config]

MaxThreads=nn

nn est le nombre maximum de connexions simultanées.

Par défaut, le nombre est 1. Cela veut dire que, si une connexion est en cours, toute nouvelle connexion attendra que la précédente soit fermée.

Attention, l'utilisation de connexions simultanées nécessite que les programmes soient conçus pour fonctionner de manière concurrente, c'est à dire gérer le partage des objets communs (fichiers, presse-papier, etc.).

Les accès concurrentiels

Pour faire des programmes LAUNCHER Office sûrs en terme d'accès concurrentiels, il faut surveiller l'accès aux ressources partagées, et avoir la certitude qu'elles ne seront pas utilisées simultanément.

Par exemple, considérons un programme A qui utilise le fichier "a.txt" par la commande DBXFER.

Ce programme est sûr en termes d'accès concurrentiels par rapport à un programme B qui utiliserait le fichier "b.txt", mais **pas par rapport à lui-même**.

Donc soit on est sûr que le programme A n'est jamais lancé simultanément, soit on résout le problème de la **ressource critique**.

Pour résoudre ce problème, les alternatives sont :

- Utiliser un nom de fichier non fixe (à déterminer par le programme). Cette solution est la meilleure car elle permettra à plusieurs instances du programme A de fonctionner en simultanéité complète.

La variable **%CONID%** permet de garantir cette unicité pour une instance de LAUNCHER Office donnée (attention aux fichiers qui seraient sur le réseau et utilisés par plusieurs postes).

- utiliser une '**section critique**' à l'aide de la commande LAUNCHER '**CRITSECT**', qui vous permet de protéger une ressource commune.

Dans ce cas, il faudra faire, juste avant d'utiliser le fichier a.txt
LNCCMD CMD(CRITSECT) PARM1('Name="ProgramA";Enter')

Et juste après:

LNCCMD CMD(CRITSECT) PARM1('Name="ProgramA";Leave')



Il est toujours préférable d'éviter les conflits (lorsque c'est possible) plutôt que d'utiliser des sections critiques.

Les sections critiques sont automatiquement libérées par la commande LNC_CLOSE.

Le presse-papier

Le presse-papier utilisé par les commandes WCOPY et WPASTE, XLCOPY et XLPASTE est déjà sécurisé.

Attention cependant à observer les règles suivantes:

- toujours faire un 'Paste' après un 'Copy' (risque de blocage d'un autre programme qui utiliserait le presse-papier)
- ne pas faire plusieurs 'Paste' pour un 'Copy' (les 'Paste' suivant n'auraient aucune garantie de contenu, car la ressource critique est libérée au premier 'Paste').

La section critique pour le presse-papier s'appelle 'CLIPBOARD'. Vous devrez l'utiliser si vous utilisez le presse-papier autrement que par les fonctions WCOPY/WPASTE/XLCOPY /XLPASTE.

Analyser la trace

Pour pouvoir lire la trace correctement il est indispensable d'afficher la colonne 'numéro de thread' qui permettra de savoir quelle connexion travaille et pouvoir ainsi retrouver le déroulement de chaque programme.

Chaque session aura un numéro de thread constant.

Cela donne par exemple:

```
00000354 Attente de connexion
00000354 Attente de connexion
00000414 LAUNCHER/400 V:LAUNCHER/400 Server version 2.0.0.0 - running
as
Service
00000414 Connexion acceptée
00000414 Début du traitement des messages
00000354 Attente de connexion
0000034C LAUNCHER/400 V:LAUNCHER/400 Server version 2.0.0.0 - running
as
Service
0000034C Connexion acceptée
0000034C Début du traitement des messages
0000034C >CRITSECT,Opt=
0000034C *ACK
00000414 >WORDOPEN,Opt=
00000414 LNC/Word Module=C:\progra~1\launcher400\LNCWORD.dll;
V.2,0,0,0
00000414 MS Word V.9.0
00000414 *ACK
```

Ici deux sessions sont ouvertes: les threads '414' et '34C'.





Gestion d'erreurs

L'interface OLE proposée par Word et Excel, permet de générer des messages d'erreurs très explicites (ouverture d'un fichier inexistant, ...).

Du côté de l'AS/400, c'est le programme LNCCMD qui génère une erreur.
Il faut se référer au JOBLOG pour en connaître la raison.

Les messages d'erreur affichés dans l'onglet Trace sont également retournés au client.

Vous pouvez les afficher grâce à la variable de retour &RESULT (voir la section [Programmation](#) de **LAUNCHER Office**).

Programmation avec LAUNCHER Office

Programmer LAUNCHER

LAUNCHER Office offre une interface de programmation destinée :

- Aux développeurs tout langage, par l'utilisation de **webservices REST**, sur Windows et Linux
- Aux développeurs **Java** utilisant des classes Java à partir d'une plateforme Windows/Linux
- Aux développeurs RPG, COBOL, CL à partir d'un **AS400**

Le programmeur dispose :

1) D'un ensemble de **programmes** constituant **l'API de base** de LAUNCHER Office quelque soit le client (Windows, Linux ou AS/400) :

- [LNCOOPEN](#), pour ouvrir une conversation avec le PC.
- [LNCCMD](#), pour envoyer des commandes au PC.
- [LNCCLOSE](#), fermer la communication avec le PC.

2) D'un ensemble de **commandes CL** de haut niveau pour les clients AS/400:

- [LNCPRTDOC](#): Composer et imprimer un document Word,
- [LNCTOXLS](#): Transférer des données vers un classeur Excel.
- [LNCXFER](#): Transférer un fichier ou une requête vers le PC.
- [LNCSHELL](#), Lancer l'exécution d'une commande ou d'un programme Windows.
- [CPYTOMDB](#): Transférer des données vers MS Access.
- [LNCSNDMAIL](#): Composer et envoyer un message électronique.

3) De classes Java :

- LNCPRTDOC
- LNCTOXLS
- DataSource

4) De webservices offrant des méthodes équivalentes à l'API (Open, Cmd et Close), et quelques autres telles que Filetransfer et Ldapauth.

Notes pour le client AS/400 :

Les programmes API de LAUNCHER sont utilisables en environnement OPM et ILE.

En environnement ILE, les procédures ont des noms identiques à leur équivalent OPM, et elles sont définies dans le programme de service "LAUNCHER".

Les programmes de LAUNCHER Office (LNCOOPEN, LNCCMD et LNCCLOSE) sont appelables par l'instruction CALL du langage de programmation : CL, RPG, COBOL, ...

Les programmes qui utilisent LAUNCHER Office peuvent tourner en batch ou en interactif.





Principe de Fonctionnement

Tout programme, quelque soit le langage et la plateforme utilisés (Windows ou Linux), peut utiliser [les webservices fournis](#).

Autre possibilité pour les client Java (Windows et Linux) et AS400 : [utiliser l'API \(LNCOPEN, LNCCMD et LNCCLOSE\)](#).

Un programme client (Java ou CL) qui utilise l'API, doit suivre les étapes suivantes pour communiquer avec un PC :

- Ouvrir une conversation avec le programme API "[LNCOPEN](#)".
- Envoyer une série de commandes à exécuter vers le PC avec "[LNCCMD](#)".
- Fermer la conversation avec le programme API "[LNCCLOSE](#)".

Le démarrage et la fermeture des applications Windows, telles que Word, Excel, Access, sont demandées par le programme client, au travers du programme API "LNCCMD".

Un client Java peut également utiliser [les classes Java fournies](#).

Un client AS400 peut utiliser [les programmes CL fournis](#).

Remarques pour le client AS/400 :

Les commandes CL de haut niveau de LAUNCHER Office (LNCPRDOC, LNCXFER, LNCTOxls, ...) prennent en charge la gestion des ouvertures et fermetures de communications, ainsi que le lancement et l'arrêt des applications Windows utilisées.

Dans le cas où seules les commandes CL sont utilisées, le programmeur n'a pas à prendre en compte les étapes décrites ci-dessus.

Les commandes LNCCMD

Les mots clés de LNCCMD

La commande **LNCCMD** nécessite entre autres les paramètres suivants :

CMD, PARM1, PARM2 et RESULT.

La valeur donnée au paramètre **CMD** du programme LNCCMD désigne l'action demandée au PC par le programme client.

Les valeurs données aux paramètres **PARM1** et **PARM2** de la commande LNCCMD dépendent de l'action demandée au PC par le mot clé CMD.

L'utilisation de ces deux paramètres est décrite pour chacun des mots clés dans la [liste alphabétique](#).

Le paramètre **RESULT** contient une valeur en retour de l'appel du programme LNCCMD.

L'utilisation de certains mots clés pour le paramètre CMD entraîne un retour de résultat.

Vous pouvez connaître la liste des mots clés pour le paramètre **CMD**, en consultant le chapitre suivant: "[Liste alphabétique](#)".

Les valeurs possibles des mots clés pour le paramètre CMD peuvent être classées en catégories :

- **Wxxxx** : Les mots commençant par W concernent une action MS Word.
- **XLxxxx & EXCELxxx** : Les mots commençant par XL ou EXCEL concernent une action MS EXCEL.
- **MAILxxxx** : Les mots commençant par MAIL concernent la messagerie.
- **OLExxxx** : Les mots commençant par OLE concernent les liens avec un objet OLE.
- **PDFxxxx**: Les mots commençant par PDF concernent la génération ou la gestion de fichiers PDF.
- **Autres** : Les autres mots concernent des actions générales sur Windows ou les applications autres.

Appel de la commande LNCCMD par un client AS/400 en CL :

```
CALL PGM(LNCCMD) PARM(&HANDLE &OPT &CMD &PARM1 &PARM2 &RESULT)
```

Ou

```
LNCCMD          CMD (EXCELOPEN)
```

Appel de la commande LNCCMD en Java :

```
test.LNCCMD("EXCELOPEN ");
```



Utilisation de LNCCMD en Java : exemple

```
package test;

import fr.aura.launcher.wrapper.LNCClientWrapper;
import fr.aura.launcher.wrapper.LNCSrvAckMsg;
import fr.aura.launcher.wrapper.LauncherException;

public class TestCMD
{
    public static void main(String[] args)
    {
        String pSvrAddr = "127.0.0.1";
        int pSvrPort = 6078;
        LNCSrvAckMsg iRetLNCCMD = null;

        LNCClientWrapper test = new LNCClientWrapper();
        test.LNCOOPEN(pSvrAddr,pSvrPort);

        try
        {
            iRetLNCCMD = test.LNCCMD("WORDOPEN",null,null,null);
        }
        catch (LauncherException e)
        {
            e.printStackTrace();
        }
        System.out.println(iRetLNCCMD.getMsg());

        Parm1 = "C:\\temp\\essai2.docx";
        String Parm2 = "visible";

        try
        {
            iRetLNCCMD = test.LNCCMD("WOPENFILE",null,Parm1,Parm2);
        }
        catch (LauncherException e)
        {
            e.printStackTrace();
        }
        System.out.println(iRetLNCCMD.getMsg());

        Parm1 = "File=C:\\temp\\testPdf_9b6_result.pdf;EmbedFonts=True";

        try
        {
            iRetLNCCMD = test.LNCCMD("PDFPRINTER",null,Parm1,null);
        }
        catch (LauncherException e)
        {
            e.printStackTrace();
        }
    }
}
```

```
    }
    System.out.println(iRetLNCCMD.getMsg());

    Parm1 = "Printer=\"LAUNCHER_PDF\"";

    try
    {
        iRetLNCCMD = test.LNCCMD("WPRINT",null,Parm1,null);
    }
    catch (LauncherException e)
    {
        e.printStackTrace();
    }
    System.out.println(iRetLNCCMD.getMsg());

    Parm1 = "Save=false";

    try
    {
        iRetLNCCMD = test.LNCCMD("WORDCLOSE",null,Parm1,null);
    }
    catch (LauncherException e)
    {
        e.printStackTrace();
    }
    System.out.println(iRetLNCCMD.getMsg());

    test.LNCCLOSE();

}

}
```

Liste des commandes

Les exemples d'utilisation des commandes sont donnés en CL, mais les commandes peuvent également être utilisées dans un programme Java.

Il faut se référer à [l'exemple sur la page précédente](#).

| |
|----------------------------|
| CHKDIR |
| CHKFILE |
| CLEANREG |
| COPYFILE |
| CREATEDIR |
| CRE |
| CRITSECT |
| CSVTOXML |
| CSVUTF8 |
| DBXFER |
| DBXPROP |
| DELETEFILE |
| DIRLIST |
| END |
| ENF |
| ENUMPRT |
| EXCELCLOSE |
| EXCELHIDE |
| EXCELOPEN |
| EXCELSHOW |
| EXE |
| FILECLOSE |
| FILEOPEN |
| FILEREAD |
| FILEWRITE |
| FTPCLOSE |
| FTPCONNECT |
| FTPDELDIR |
| FTPDELFILE |
| FTPGETDIR |

| |
|----------------------------|
| FTPGETFILE |
| FTPLISTDIR |
| FTPMKDIR |
| FTPPUTDIR |
| FTPPUTFILE |
| GETREGE |
| GETSYSINFO |
| GRAPHSEND |
| IFSPUT |
| LISTSCAN |
| LNCDBGET |
| MAILATT |
| MAILBODYF |
| MAILCC |
| MAILEND |
| MAILLATT |
| MAILMSG |
| MAILMCHG |
| MAILMDLT |
| MAILPREP |
| MAILPRTY |
| MAILREPORT |
| MAILSEND |
| MAILSMTP |
| MAILSUBJ |
| MAILTEXT |
| MAILTO |
| MAIXATT |
| MENU |
| MENUWAIT |
| MIN |
| MOVEFILE |
| MSGBOX |
| NCL |
| NLN |
| NOP |



| |
|----------------------------|
| OLECALL |
| OLECREATE |
| OLERELEASE |
| PARSEXML |
| PDFALLXL |
| PDFCONCAT |
| PDFCOUNT |
| PDFENCRYPT |
| PDFEXTRACT |
| PDFFACTX |
| PDFFRIM |
| PDFFRXL |
| PDFMERGE |
| PDFORDERX |
| PDFOVER |
| PDFPRINT |
| PDFPRINTER |
| PDFSEARCH |
| PDFSIGN |
| PDFSPLIT |
| PDFTOIM |
| PRINTPRNF |
| PROPERTY |
| RCL |
| REMOVEDIR |
| RSTPRINTER |
| SAISI |
| SAISWAIT |
| SCAN |
| SELECTFILE |
| SETPRINTER |
| SHELL |
| STO |
| VAL |
| WADDINS |
| WBOOKMADD |



| |
|----------------------------|
| WBOOKMARK |
| WBOOKMDEL |
| WCHDIR |
| WCLOSEFILE |
| WCOPY |
| WDOCUMENT |
| WDPROTECT |
| WEXEMACRO |
| WFIELDS |
| WFINDTEXT |
| WFORMFIELD |
| WGETHPOS |
| WGETPAGE |
| WGETPROP |
| WGETTEXT |
| WGETVPOS |
| WGOTOTBL |
| WINSERTOBJ |
| WINSERTBRK |
| WINSERTCOL |
| WINSERTF |
| WINSERTIMG |
| WINSERTROW |
| WINSERTTBL |
| WMAILMERGE |
| WMAXIMIZE |
| WMENU |
| WMERGECELL |
| WMETHOD |
| WMINIMIZE |
| WMOVE |
| WMOVERIGHT |
| WNEWFILE |
| WOPENFILE |
| WORDCLOSE |
| WORDHIDE |



| |
|----------------------------|
| WORDOPEN |
| WORDSHOW |
| WORDWAIT |
| WPAGEBRK |
| WPASTE |
| WPRINT |
| WPROTECT |
| WREFVAR |
| WREPLACE |
| WSAVE |
| WSAVEAS |
| WSELECT |
| WSELROW |
| WSETLINE |
| WSETVAR |
| WSETPASSWD |
| WSETPROP |
| WSPLITCELL |
| WTYPETEXT |
| XLADDINS |
| XLADDSHEET |
| XLBOLD |
| XLCALCULAT |
| XLCELLNAME |
| XLCELLS |
| XLCLOSFILE |
| XLCOPY |
| XLCOPYSH |
| XLDELSHEET |
| XLDISADD |
| XLDPROTECT |
| XLDRAWGR |
| XLENAADD |
| XLEXEMACRO |
| XLEXISTSH |
| XLFIND |



| |
|----------------------------|
| XLFORMULA |
| XLGETFILE |
| XLGETPOS |
| XLGETPROP |
| XLGETVALUE |
| XLGOTOCELL |
| XLGOTOSH |
| XLINSERT |
| XLINSERCL |
| XLINSERTRW |
| XLITALIC |
| XLMAXIMIZE |
| XLMENU |
| XLMETHOD |
| XLMINIMIZE |
| XLOPENFILE |
| XLPAGEBRK |
| XLPASTE |
| XLPRINT |
| XLPROTECT |
| XLREPLACE |
| XLRESIZE |
| XLSAVEAS |
| XLSETLINE |
| XLSETPROP |
| XLSETTEXT |
| XLSETVALUE |
| XLSHNAME |
| XLSORT |
| XLSUBTOTAL |
| XLUNDERLN |
| XLWAIT |

Commande *CHKDIR*

**Permet de vérifier l'existence d'un répertoire Windows.
248 caractères maximum pour le chemin du répertoire.**

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('CHKDIR')
CHGVAR          VAR(&PARM1) VALUE('
Dir="Chemin complet du répertoire à vérifier"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | Dir : Chemin complet du répertoire à vérifier. |
| RESULT | TRUE : si le répertoire existe. FALSE : si le répertoire n'existe pas. |

Exemple

```

LNCCMDR        CMD (CHKDIR) +
                PARM1 ('DIR="C:\AURADATA\SUPPORT\TKM"') +
                RESULT (&RES)
LNCCMD         CMD (NOP) PARM1 (&RES)
    
```

*Commande **CHKFILE***

Permet de tester la présence d'un fichier ou d'un répertoire sur le PC et les droits sur un fichier.

Utilisez [CHKDIR](#) pour tester un répertoire réseau.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('CHKFILE')
CHGVAR          VAR(&PARM1) VALUE('
                "Fichier"
                [;Read]
                [;Write]
                [;Wait=nombre de secondes]
                [;GetInfo]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
                &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>Chemin complet du fichier ou du répertoire.</p> <p>Read : L'option "Read" est spécifiée pour tester si on a le droit de lire le fichier.</p> <p>Write : L'option "Write" est spécifiée pour tester si on a le droit d'écrire dans le fichier.</p> <p>Wait=nnn; nnn est un nombre de secondes. Délai d'attente pour que le fichier soit présent et disponible.</p> <p>GetInfo=True; Pour recevoir dans RESULT les informations sur le fichier.</p> <p>Position 7 : AAAAMMJJ : Date dernière modification.</p> <p>Position 16 : HMMSS.mmm : Heure dernière modification.</p> <p>Position 27 : TTTTTTTTTTTT : Taille en octet (12 chiffres).</p> <p>Position 40 : D Si le chemin désigne un répertoire.</p> |
| RESULT | <p>En retour, le paramètre &RESULT contient une des valeurs suivantes :</p> <p>Position 1 : TRUE / FALSE / DENIED / SHV</p> <p>Position 7 : Informations sur le fichier si l'option GetInfo est choisie.</p> <p>TRUE : Le fichier existe, et on a les droits demandés.</p> <p>FALSE : Le fichier n'existe pas.</p> |



DENIED : Le fichier existe, mais on n'a pas les droits de lecture ou d'écriture demandés.

SHV : Le fichier existe, mais il est en cours d'utilisation.

Exemple

```
CHGVAR VAR(&CMD) VALUE('CHKFILE')
CHGVAR VAR(&PARM1) VALUE('%LNCDIR%\SAMPLES\LNCMENU.DOC';READ;+
WRITE')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Autre exemple

```
LNCCMDR CMD(CHKFILE) PARM1(' '*TCAT &OUTDIR *TCAT +
'\ '*TCAT &NOMLOT *TCAT +
'.pdf";Getinfo=true') RESULT(&RESULT)
```

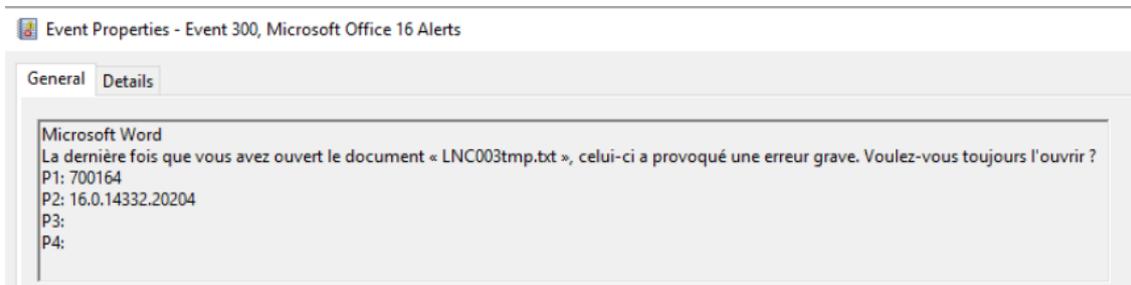
Commande CLEANREG

Quand un publipostage (commande WMAILMERGE) termine en erreur avec Launcher, il se peut que Microsoft Word désactive le fichier temporaire de données (%TEMP%\LNC00xtmp.txt). Dès lors tous les publipostages suivants ne marcheront pas.

Dans la trace Launcher, vous aurez le message suivant :

**REJ800706be:The remote procedure call failed*

De plus, dans l'observateur d'évènements Windows, vous aurez cette information :



La commande CLEANREG permet de réactiver les fichiers temporaires de données.

A noter que ce traitement est fait automatiquement au démarrage du service Launcher. Cette commande est donc utile seulement lorsque Launcher est lancé en tant qu'application.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('CLEANREG')
CHGVAR          VAR(&PARM1) VALUE('
                Appli="Word"
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
                &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | Appli : Pour le moment, la seule valeur possible est : Word . |

Exemple



```
CHGVAR VAR(&CMD) VALUE('CLEANREG')
CHGVAR VAR(&PARM1) VALUE('Appli="Word"')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande COPYFILE

Permet de copier un fichier d'un répertoire vers un autre sur le PC.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('COPYFILE')
CHGVAR          VAR(&PARM1) VALUE('
File="Fichier à copier"
NewFile="Fichier copié"
[;Replace=true/false]
[;ReadOnly=true/false]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>File : Chemin complet et nom du fichier à copier.</p> <p>NewFile : Chemin complet et nom du fichier copié.</p> <p>Replace = true, pour remplacer automatiquement le fichier copié s'il existe déjà. Valeur par défaut : false.</p> <p>ReadOnly=true, pour spécifier l'attribut READ-ONLY sur le fichier copié. Valeur par défaut : false.</p> |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```

LNCCMD CMD(COPYFILE) PARM1('File="C:\temp\SBOOK_template.xls";
+ NewFile="C:\temp\TEST\copy2.xls";replace=true')
    
```

Commande CREATEDIR

Permet de créer un répertoire Windows.

Plusieurs répertoires (intermédiaires et final) peuvent être créés en une seule fois, si l'option FullPath est égale à true.

Seulement le dernier répertoire du chemin sera créé si FullPath est égale à false (ou absente).

Les répertoires créés, hériteront des paramètres de sécurité du répertoire parent.

248 caractères maximum pour le chemin du répertoire.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('CREATEDIR')
CHGVAR          VAR(&PARM1) VALUE('
Dir="Chemin complet du répertoire à créer"
[;FullPath=true/false]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | Dir : Chemin complet du répertoire à créer. |
| | FullPath : Mettre à true (par défaut : false) pour créer les répertoires intermédiaires et le répertoire final, du chemin spécifié via le paramètre Dir. |
| RESULT | Contient éventuellement le message d'erreur. |

Exemples



```
LNCCMD      CMD(CREATEDIR) +  
            PARM1('Dir="C:\temp\test\AURA\result"')
```

Le répertoire final "result" sera créé si "C:\temp\test\AURA" existe déjà.

```
LNCCMD CMD(CREATEDIR) PARM1('Dir="C:\A\DV\KAT";FullPath=true')
```

Les répertoires "DV" et "KAT" seront créés avec FullPath=true, car "C:\A" existe déjà.

Commande CRF

Permet la création d'un fichier vide sur le PC.

N'oubliez pas de finaliser la création du fichier avec la commande [ENE](#).

Ce fichier peut être ensuite utilisé comme fichier de données sur le PC.

Le fichier peut être rempli par programmation avec les commandes NCL, NLN et VAL.

Ou on peut utiliser la commande [DBFXFER](#) pour le transfert d'un fichier de base de données.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('CRF')
CHGVAR          VAR (&PARM1) VALUE ('Fichier')
CHGVAR          VAR (&PARM2) VALUE (' ')

CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin complet du fichier à créer. Le fichier est créé vide. S'il existe déjà, il est effacé. |

Exemple 1

```
LNCOPEN

LNCCMD          CMD (CRF) PARM1 ('C:\A\fusion2.txt')

LNCCMD          CMD (NCL) PARM1 ('Nom')

LNCCMD          CMD (NCL) PARM1 ('Prenom')

LNCCMD          CMD (NCL) PARM1 ('Identifiant')

LNCCMD          CMD (NLN)

LNCCMD          CMD (VAL) PARM1 ('Nom') PARM2 ('Redfall')

LNCCMD          CMD (VAL) PARM1 ('Prenom') PARM2 ('Robert')
```

```
LNCCMD      CMD (VAL) PARM1 ('Identifiant') PARM2 ('1')
```

```
LNCCMD      CMD (ENF)
```

```
LNCCLOSE
```

Dans cet exemple, le fichier 'fusion2.txt' est créé dans le répertoire "C:\A".

Exemple 2

```
LNCOOPEN
```

```
LNCCMD      CMD (CRF) PARM1 ('C:\A\fusion.txt')
```

```
LNCCMD      CMD (ENF)
```

```
LNCCMD      CMD (DBXFER) PARM1 ('FILE="C:\A\fusion.txt"') +  
              PARM2 ('SQL:select CUST_ID as Identifiant, +  
              FIRSTNAME as Prenom, LASTNAME as Nom from +  
              SP_CUST')
```

```
LNCCLOSE
```

Voir aussi

- [ENF](#)
- [NCL](#)
- [NLN](#)
- [VAL](#)
- [DBXFER](#)

Commande CRITSECT

Permet de gérer les conflits d'accès à des ressources du PC, dans le cas où le serveur LAUNCHER Office du PC est configuré pour accepter plusieurs **connexions en simultanées**.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('CRITSECT')
CHGVAR          VAR(&PARM1) VALUE('Name="Nom" [;Enter] | [;Leave]')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                               &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Name est un nom symbolique de ressource. Enter indique que l'application entre dans une phase critique, qui se terminera par une exécution de CRITSECT, sur la même ressource, avec l'option Leave. Leave indique que l'application quitte une phase critique. Une autre application peut y entrer à son tour. |
| Parm2 | |

Consultez le chapitre sur la [configuration multi connexions](#) du serveur LAUNCHER Office.

Quand une application doit utiliser une ressource partagée sur un PC, telle que le presse papier, ou un fichier de travail, elle doit entrer dans une phase critique et en ressortir quand l'utilisation de cette ressource est terminée.

L'application restera en attente si une autre application est déjà dans cette même phase critique, jusqu'à la libération.

Le nom de la ressource est virtuel. Il est fixé par l'application.

Exemple

```
CHGVAR VAR(&CMD) VALUE('CRITSECT')
CHGVAR VAR(&PARM1) VALUE('Name="PhaseCrit";Enter')
CALL PGM(LNCCMD) PARM(&HANDLE &OPT &PARM1 &PARM2 &RESULT)

/* Copier vers le presse papier */
CHGVAR VAR(&CMD) VALUE('WCOPY')
CALL PGM(LNCCMD) PARM(&HANDLE &OPT &PARM1 &PARM2 &RESULT)

/* Coller le contenu du presse papier */
```

```
CHGVAR VAR(&CMD) VALUE('WPASTE')
CALL PGM(LNCCMD) PARM(&HANDLE &OPT &PARM1 &PARM2 &RESULT)

/* On peut maintenant sortir */
CHGVAR VAR(&CMD) VALUE('CRITSECT')
CHGVAR VAR(&PARM1) VALUE('Name="PhaseCrit";Leave')
CALL PGM(LNCCMD) PARM(&HANDLE &OPT &PARM1 &PARM2 &RESULT)
```

Commande CSVTOXML

Permet de générer un fichier XML à partir d'un fichier CSV (séparateur : point-virgule).

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('CSVTOXML')
CHGVAR          VAR(&PARM1) VALUE('
                CSV="Chemin du fichier CSV";
                XML="Chemin du fichier XML résultat";
                ElemName="Nom de la balise"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>CSV : Contient le chemin complet du fichier CSV (séparateur point-virgule).</p> <p>XML : Contient le chemin complet du fichier XML.</p> <p>ElemName : Indique le nom de chaque balise correspondant à chaque enregistrement du fichier CSV.</p> |
| Parm2 | |

Exemple

```

PGM

DCL          VAR(&SVRADDR) TYPE(*CHAR) LEN(30) VALUE(*DEV)
DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&CCSID) TYPE(*CHAR) LEN(10)

DCL          VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)

CHGVAR      VAR(&HANDLE) VALUE('*ONLY')
CHGVAR      VAR(&CCSID) VALUE('*JOB')

CALL        PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
    
```

```

CHGVAR      VAR (&CMD) VALUE ('CSVTOXML')
CHGVAR      VAR (&PARM1) +
            VALUE ('CSV="C:\A\csv.txt";XML="C:\A\csv_xml+.xml";ElemName="SP_CUST"')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
            &PARM2 &RESULT)

LNCCLOSE

ENDPGM

```

Avec ce fichier CSV :

```

"CUST_ID";"COMPANY";"FIRSTNAME";"LASTNAME";"CIVIL";"ADDRESS";"ADDR2";"CITY";"STATE";"ZIP";"COUNTRY";"PHONE";"FAX"
"1231";"Unisco";"George";"Weathers";"1";"PO Box Z-547";"";"Freeport";"";"Bahamas";"809-555-3915";"809-555-4958"
"1351";"Sight Diver";"Phyllis";"Spoonier";"1";"1 Neptune Lane";"";"Kato Paphos";"";"Cyprus";"357-6-876708";"357-6-870943"

```

Le fichier XML généré, est le suivant :

```

<?xml version="1.0" standalone="yes"?>
<DocumentElement>
  <SP_CUST>
    <CUST_ID>"1231"</CUST_ID>
    <COMPANY>"Unisco"</COMPANY>
    <FIRSTNAME>"George"</FIRSTNAME>
    <LASTNAME>"Weathers"</LASTNAME>
    <CIVIL>"1"</CIVIL>
    <ADDRESS>"PO Box Z-547"</ADDRESS>
    <ADDR2>""</ADDR2>
    <CITY>"Freeport"</CITY>
    <STATE>""</STATE>
    <ZIP>""</ZIP>
    <COUNTRY>"Bahamas"</COUNTRY>
    <PHONE>"809-555-3915"</PHONE>
    <FAX>"809-555-4958"</FAX>
  </SP_CUST>
  <SP_CUST>
    <CUST_ID>"1351"</CUST_ID>
    <COMPANY>"Sight Diver"</COMPANY>
    <FIRSTNAME>"Phyllis"</FIRSTNAME>
    <LASTNAME>"Spoonier"</LASTNAME>
    <CIVIL>"1"</CIVIL>
    <ADDRESS>"1 Neptune Lane"</ADDRESS>
    <ADDR2>""</ADDR2>
    <CITY>"Kato Paphos"</CITY>
    <STATE>""</STATE>
    <ZIP>""</ZIP>
    <COUNTRY>"Cyprus"</COUNTRY>
    <PHONE>"357-6-876708"</PHONE>
    <FAX>"357-6-870943"</FAX>
  </SP_CUST>
</DocumentElement>

```

</SP_CUST>
</DocumentElement>



Commande CSVUTF8

Permet de convertir un fichier texte issu de la commande [DBFXFER](#), au format UTF-8.
 Par défaut, le mode de transfert des données est l'Unicode avec Launcher (UTF-16LE).

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('CSVTOXML')
CHGVAR          VAR(&PARM1) VALUE('
                CSVIn="Chemin du fichier texte en entrée"
                ;CSVUTF8="Chemin du fichier texte UTF-8"
                [;CodeIn="Encodage du fichier texte en entrée"]')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>CSVIn : Contient le chemin complet du fichier texte issu de la commande DBFXFER.</p> <p>CSVUTF8 : Contient le chemin complet du fichier texte converti en UTF-8.</p> <p>CodeIn : Optionnel. Peut prendre les valeurs "UTF-16LE" ou "WINDOWS-1252". Si ce paramètre est omis, sa valeur par défaut est : "UTF-16LE". En effet, le mode de transfert des données par défaut, est l'Unicode ("UTF-16LE") avec Launcher.</p> |
| Parm2 | |

Exemple

```
PGM

DCL          VAR(&SVRADDR) TYPE(*CHAR) LEN(30) VALUE(*DEV)
DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&CCSID) TYPE(*CHAR) LEN(10)
```

```
DCL          VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)

CHGVAR      VAR(&HANDLE) VALUE('*ONLY')
CHGVAR      VAR(&CCSID) VALUE('*JOB')

CALL        PGM(LNCOOPEN) PARM(&HANDLE &SVRADDR &CCSID)

CHGVAR      VAR(&CMD) VALUE('CSVUTF8')
CHGVAR      VAR(&PARM1) +
            VALUE('CSVIn="C:\A\csv.txt";CSVUTF8="C:\A\csv_utf8+
            .txt"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
            &PARM2 &RESULT)

LNCCLOSE

ENDPGM
```

Commande DBFXFER

Transfère un fichier de base de données AS/400, ou le résultat d'une requête SQL, ou un document de l'IFS, vers un PC.

Le résultat du transfert d'un fichier base de données, ou d'une requête SQL sera en format texte sur le PC.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('DBFXFER')
CHGVAR      VAR(&PARM1) VALUE('
File="Fichier PC cible"
[;UNICODE=True/False]
[;Fixed=True/False]
[;FixedNL=True/False]
[;ColHdg=True/False]
[;Delimiter="délimiteur"]
[;DecimalPt="point décimal"]
[;QuoteText=True/False]
[;DescFile=True/False]
')
CHGVAR      VAR(&PARM2) VALUE('
MAXREC=Nombre maximum d'enregistrements ;
COLHDG=LNAME / COLHDG / SNAME ;
<Fichier Base de données / Document IFS / Requête SQL>
')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2+ &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Parm1 contient des options qui seront interprétées par LAUNCHER Office sur le PC.</p> <p>File=Chemin complet vers le fichier cible à créer sur le PC. Si le mot "FILE=" est omis, alors tout le paramètre <i>Parm1</i> représente le chemin vers le fichier cible.</p> <p>UNICODE=True / False (<u>Défaut=False</u>). Par défaut, les données sont stockées en format 8 bits par caractères. Au format UNICODE les caractères sont codés sur 16 Bits.</p> <p>Fixed=True / False (<u>Défaut=False</u>). Par défaut, le fichier sera au format CSV. Les colonnes sont séparées par un point virgule. Si Fixed est vrai, les colonnes seront de taille fixe, sans séparateur</p> |



de colonnes ou de lignes.

FixedNL

Comme pour **Fixed**, les colonnes seront de taille fixe, mais les lignes seront séparées par un "retour ligne".

ColHdg=True / False (Défaut=True). Par défaut, la première ligne du fichier résultat contient les noms des colonnes. Si **ColHdg** est faux, les données commenceront dès la première ligne.

Delimiter : Le séparateur de colonnes est le point virgule (;) par défaut. Ce séparateur peut être changé ici.

Exemple : **Delimiter=","**

Le délimiteur peut être totalement supprimé par :

Delimiter=""

Si le fichier ne comporte qu'un seul champ, pour des raisons de compatibilité avec MS Office, un délimiteur est ajouté à la fin de chaque ligne.

DecimalPt : Le point décimal par défaut sera celui défini dans l'environnement Windows du PC. Il peut être changé ici.

Exemple: **DecimalPt=","**

QuoteText=True / False (Défaut=True). Les valeurs des colonnes sont encadrées par des guillemets, sauf si **QuoteText** est faux.

DescFile=True / False (Défaut=True). Un fichier de description (*.LFD) est généré par LAUNCHER Office dans le même répertoire que le fichier de destination. Ce fichier de description est utilisé par LAUNCHER Office lors d'une exportation vers Excel. Pour économiser des ressources, la génération de ce fichier de description peut être évitée en donnant la valeur faux à **DescFile**.

Parm2

Parm2 contient des options qui seront interprétées par LAUNCHER sur l'AS/400.

Les options doivent figurer en début, séparées par point virgule (;). La valeur de Parm2 se termine par le nom de la source de données AS/400 à transférer.

MAXREC=*Nombre*; Cette option permet de plafonner le transfert à un nombre maximum d'enregistrement. Par défaut, il n'y a aucun plafond.

COLHDG= *LNAME* / *COLHDG* / *SNAME*

Les entêtes des colonnes sur le fichier PC seront constituées :

Si COLHDG=LNAME : par les nom longs ou Alias.

Si COLHDG=COLHDG : par les entêtes de colonnes.

Si COLHDG=SNAME : par les noms courts (Défaut).

Fichier base de données : Donnez le nom du fichier, avec éventuellement la librairie et le membre.
La syntaxe d'un nom qualifié est : LIBRAIRIE/FICHER(MEMBRE)
La librairie et le membre sont optionnels.

Document de l'IFS : Donnez le chemin complet vers le document dans les dossiers, précédé du mot « **DOCUMENT:**»
Exemple : *'DOCUMENT:/qdfs/dossierdoc/lettre.pdf'*

Résultat d'une requête SQL : Le mot « **SQL:**» doit précéder le texte de la requête SQL.
Exemple : *'SQL:select * from customer where dept="75"'*
Si un fichier doit être qualifié dans le texte de la requête, le nom de librairie et le nom de fichier doivent être séparés par un caractère point, selon la syntaxe SQL.
Exemple : *'SQL:select * from mylib.myfile'*

RESULT En retour &RESULT contient le nombre de records (9 chiffres).

Cette fonction de transfert de données ne nécessite aucun autre logiciel que LAUNCHER Office sur le PC.

Pour le transfert d'un document de l'IFS, il n'est pas utile de connecter un disque virtuel du PC à l'IFS.

Remarque : Un fichier .lfd est également créé.

Exemple

```
CHGVAR            VAR(&CMD) VALUE ('DBFXFER')
CHGVAR            VAR(&PARM1) VALUE ('file="%LNCDIR%\fusion.txt ')
CHGVAR            VAR(&PARM2) VALUE ('COLHDG=LNAME;LAUNCHER/SP_CUST')
CHGVAR            VAR(&CMD) VALUE ('DBFXFER')
CHGVAR            VAR(&PARM1) VALUE ('%TEMP%\readme.txt')
CHGVAR            VAR(&PARM2) VALUE ('DOCUMENT:/home/qsecofr/vim-
                  6.3/readme.txt')

CHGVAR            VAR(&CMD) VALUE ('DBFXFER')
CHGVAR            VAR(&PARM1) VALUE ('%TEMP%\trf.txt ')
CHGVAR            VAR(&PARM2) VALUE ('MAXREC=100;SQL:select * from customer')
```

Commande DBXPROP

Sert à contrôler le transfert d'un fichier de l'AS/400 vers le PC et les options associées.

Elle doit **impérativement** être utilisée avant la [commande DBXFER](#).

Par défaut la commande DBXFER crée un fichier ASCII avec les propriétés suivantes :

- séparateur de champ ";" (point virgule)
- séparateur d'enregistrement (saut de ligne)
- tous les champs entourés de guillemets et les entêtes de colonnes en première ligne.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('DBXPROP')
CHGVAR          VAR(&PARM1) VALUE('Propriété')
CHGVAR          VAR(&PARM2) VALUE('Valeur')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2 &RESULT)
```

Paramètres

| Propriété (Parm1) | Valeur (Parm2) | Défaut | Description |
|-------------------|---------------------------------|--------|---|
| FLDSEP | Chaîne | ; | définir le séparateur de champs |
| COLHDG | OFF SNAME LNAME COLHDG | SNAME | OFF = Ne pas avoir les entêtes de colonne. SNAME = Le nom court de zone est utilisé comme nom de colonne. LNAME = Le nom long de zone, ou alias, est utilisé. COLHDG = Le "Column Heading" de la zone est utilisé. |
| NUMPREFSUF | ON/OFF | OFF | Ne pas entourer les valeurs numériques de guillemets |
| FLDPREF | chaîne | " | Caractère de début de champ. |
| FLDSUF | chaîne | " | Caractère ou chaîne de fin de champ. |
| DESCFILE | ON / OFF | ON | Génère. le fichier de description (*.lfd). |
| UNICODE | ON / OFF | OFF | Produire un fichier texte au format UNICODE. |
| DBLQUOTE | ON / OFF | ON | ON : Les doubles quotes (guillemets) peuvent être doublées dans les valeurs des zones. OFF : non doublés |

Exemple

```
CHGVAR      VAR (&CMD)  VALUE ('DBXPROP')  
CHGVAR      VAR (&PARM1) VALUE ('DESCFILE')  
CHGVAR      VAR (&PARM2) VALUE ('OFF')
```

La prochaine commande DBFXFER ne créera pas de fichier de description (.lfd) associé au fichier de données (.txt).

Autre exemple

Désactiver le transfert en Unicode :

```
LNCCMD      CMD (DBXPROP) PARM1 ('UNICODE') PARM2 ('OFF')
```

Voir aussi

- [DBFXFER](#)

Commande *DELETEFILE*

Permet de supprimer un fichier d'un répertoire sur le PC.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('DELETEFILE')
CHGVAR          VAR(&PARM1) VALUE('
File="Fichier à supprimer"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | File : Chemin complet et nom du fichier à supprimer. |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```
LNCCMD CMD(DELETEFILE) PARM1('File="C:\temp\res.pdf"')
```

Commande DIRLIST

Permet de consulter la liste des fichiers contenus dans un répertoire.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('DIRLIST')
CHGVAR          VAR(&PARM1) VALUE('
[Pattern="Fichiers à lister"]
[ ;Path="Chemin vers le répertoire"]
[ ;First= True / False ]
[ ;Close= True / False ]
[ ;SubDir= True / False ]
[ ;Hidden= True / False ]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>Pattern= : Permet d'indiquer le radical des fichiers à lister. Exemple : Pattern="*.doc"</p> <p>Le mot clé « Pattern » peut aussi contenir le chemin complet vers le répertoire. Exemple : Pattern =\\FileServer\Docs*.doc</p> <p>Path= : Indique le chemin vers le répertoire, si celui ci n'a pas été donné par le mot clé « Pattern ». Exemple : Path="\\FileServer\Docs"; Pattern="*.doc"</p> <p>First=True Indique que l'on veut démarrer le listage des fichiers au début du répertoire. Si First est faux, la commande retournera l'entrée suivante dans le répertoire.</p> <p>Close=True Termine le parcours du répertoire.</p> <p>SubDir=True Indique qu'on ne veut lister que les sous répertoires du répertoire indiqué, et non les fichiers. Si « SubDir » est faux, seul les fichiers sont listés.</p> <p>Hidden=True Indique que l'on veut inclure les fichiers cachés dans le listage.</p> |

RESULT

A chaque appel, un nom de fichier est retourné dans le paramètre **&RESULT**.

Si la liste des fichiers a été retournée en totalité, &RESULT est vide.

Chaque fichier est retourné selon la structure suivante :

- Positions 1 à 256 : Nom du fichier.
- Position 257 : '**N**'. La lettre 'N' est présente si une entrée est présente.
- Position 258 : '**A**'. Si le fichier a l'attribut 'Archive'.
- Position 259 : '**H**'. Si le fichier est caché.
- Position 260 : '**R**'. Si le fichier est en lecture seule.
- Position 261 : '**D**'. Si l'entrée est un sous répertoire.
- Position 262 : '**S**'. Si c'est un fichier système.

Exemple

```
CHGVAR      VAR(&CMD) VALUE('DIRLIST')
CHGVAR      VAR(&PARM1) VALUE('Pattern="*.xls";Path="N:\DOCS"')
CHGVAR      VAR(&PARM2) VALUE('First=True')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande *END*

Permet de se déconnecter (fin de la session de commandes).

Word et Excel restent actifs s'ils sont visibles.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('END')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [LNCOPEN](#)
- [LNCCLOSE](#)

Commande ENF

Permet de terminer la création d'un fichier de fusion.

Le fichier sera alors sauvegardé selon le nom spécifié lors de l'appel à la commande [CRF](#).

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('ENF')
CHGVAR          VAR(&PARM1) VALUE(' ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [CRF](#)
- [NCL](#)
- [NLN](#)
- [VAL](#)

Commande *ENUMPRT*

Liste les imprimantes installées sur le PC Windows.

Chaque appel à ENUMPRT retourne dans &RESULT la description d'une imprimante.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('ENUMPRT')
CHGVAR          VAR(&PARM1) VALUE('
[ First=True / False ]
[; NameOnly=True / False ]
[; Close=True ; False ] ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>First : Vrai pour initialiser le listage, et recevoir la description de la première imprimante. Si First est faux, &RESULT contiendra la description de l'imprimante suivante.</p> <p>NameOnly : Si NameOnly est vrai, la variable &RESULT ne contiendra que le nom de l'imprimante. Sinon, &RESULT contient une description sous la forme : Printer="Nom de l'imprimante";Server="Nom du serveur"</p> <p>Close : Vrai pour terminer le listage des imprimantes.</p> |
| RESULT | <p>En sortie, Le paramètre &RESULT contient :</p> <ul style="list-style-type: none"> • *NONE si la fin de la liste des imprimantes est atteinte. • La description d'une imprimante, selon un format dépendant de la valeur du mot clé NameOnly. |

Exemple

```

PGM
DCL          VAR(&RES) TYPE(*CHAR) LEN(2000)
    
```

```
DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL          VAR(&RES2) TYPE(*CHAR) LEN(2000)
DCL          VAR(&HDL) TYPE(*CHAR) LEN(100) VALUE('*ONLY ')
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&PARM2) TYPE(*CHAR) LEN(1024)

LNCOOPEN

CHGVAR      VAR(&PARM1) VALUE('FIRST=TRUE')
CALL        PGM(LNCCMD) PARM(&HDL 'ENUMPRT' &OPT &PARM1 +
                          &PARM2 &RES)

CHGVAR      VAR(&PARM1) VALUE('FIRST=FALSE')

DOWHILE     COND(&RES *NE '*NONE')

CALL        PGM(LNCCMD) PARM(&HDL 'ENUMPRT' &OPT &PARM1 +
                          &PARM2 &RES)

ENDDO

LNCCLOSE

ENDPGM
```

Commande *EXCELCLOSE*

Ferme l'application Microsoft EXCEL.

Les messages standard EXCEL de sortie s'afficheront si l'intervention de l'utilisateur est nécessaire (fichier modifié non sauvegardé, ...).

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('EXCELCLOSE')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Exemple

```
LNCCMD      CMD (EXCELCLOSE)
```

Voir aussi

- [EXCELOPEN](#)

Commande *EXCELHIDE*

La commande **EXCELHIDE** cache l'instance utilisée.
L'utilisateur ne voit plus le document, et les opérations faites par Launcher sur la feuille seront plus rapides.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('EXCELHIDE')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Exemple

```
LNCCMD          CMD (EXCELHIDE)
```

Voir aussi

- [EXCELSHOW](#)

Commande EXCELOPEN

Ouvre Microsoft EXCEL.

Si aucun paramètre n'est utilisé, seule l'application Microsoft Excel est lancée.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('EXCELOPEN')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Nom du fichier à ouvrir. Peut être vide pour la création d'un nouveau document ou contenir le nom d'un modèle si on spécifie NEW en paramètre 2. |
| Parm2 | Visible : pour rendre Excel visible True/False (Valeur par défaut : false) New = True/False (Valeur par défaut : false) Si Parm1 est vide, et New=true, cela permet la création d'un nouveau document. |

Exemple

```
LNCCMD          CMD (EXCELOPEN)
```

Voir aussi

- [EXCELCLOSE](#)
- [WORDOPEN](#)

Commande *EXCELSHOW*

La commande EXCELSHOW permet de rendre visible l'instance utilisée, pour permettre à l'utilisateur de consulter ou modifier le document en cours.

A l'ouverture de Excel par la commande EXCELOPEN, l'instance de Excel est cachée par défaut.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('EXCELSHOW')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Exemple

```
LNCCMD      CMD (EXCELSHOW)
```

Voir aussi

- [EXCELHIDE](#)

Commande EXE

Permet l'exécution d'une application PC.

Cette commande est suivie d'une option (0, 1 ou 2) indiquant au démon l'événement qu'il devra attendre avant de renvoyer le message au serveur.

- **Option 0** : Le démon attend la fin de l'exécution de l'application pour renvoyer le message au serveur,
- **Option 1** : Le démon renvoie dès le lancement de l'application le message au serveur,
- **Option 2** : Le démon envoie le message au serveur lorsqu'un événement est déclenché par l'application.

Si vous développez vous-même l'application, il est possible d'utiliser la bibliothèque LNCSmsg et les fonctions suivantes :

[LNCSMessage](#)

[LNCSRcvMessage](#)

[LNCSSetPostMessage](#)

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('EXE')
CHGVAR      VAR(&OPT) VALUE('0 | 1 | 2')
CHGVAR      VAR(&PARM1) VALUE('Application [Params] ')
CHGVAR      VAR(&PARM2) VALUE('[Params] ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Ligne de commande de l'application et éventuellement ses paramètres. |
| Parm2 | Peut contenir les paramètres à passer à l'application. |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('EXE')
CHGVAR      VAR(&OPT) VALUE('0')
CHGVAR      VAR(&PARM1) VALUE('c:\chemin\application.exe')
```

```
CHGVAR      VAR(&PARM2) VALUE('paramètres')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Autre exemple

```
LNCCMD      CMD(EXE) PARM1('%TEMP%\scan.exe') +
            PARM2('"C:\temp\test2.pdf" /contraste=10 +
            /brillance=10 /resolution=300') OPT('2')
```

Voir aussi

- [SHELL](#)

Commande FILECLOSE

Ferme le fichier texte précédemment ouvert par FILEOPEN.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('FILECLOSE')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Exemple

Cet exemple ouvre le fichier « C:\temp\texte.txt », en écriture, et efface son contenu actuel.

Le programme écrit ensuite un texte dans le fichier,
Puis ferme le fichier.

```
CHGVAR      VAR(&CMD) VALUE('FILEOPEN')
CHGVAR      VAR(&PARM1) VALUE('File="C:\Temp\texte.txt";Truncate=True')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

CHGVAR      VAR(&CMD) VALUE('FILEWRITE')
CHGVAR      VAR(&PARM1) VALUE('Texte à écrire dans le fichier')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

CHGVAR      VAR(&CMD) VALUE('FILECLOSE')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Autre exemple

```
LNCCMD      CMD(FILECLOSE)
```

Voir aussi

- [FILEOPEN](#)
- [FILEWRITE](#)

Commande FILEOPEN

Ouvre un fichier de type texte sur le PC .

Si le fichier n'existe pas, il est créé.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('FILEOPEN')
CHGVAR          VAR(&PARM1) VALUE('File="nom du fichier" [;UNICODE]
[ ;Truncate] [ ;InsertNL]')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>File : Chemin du fichier à ouvrir.</p> <p>UNICODE : True / False (Défaut=False) Préciser "UNICODE=True" pour générer un fichier en format UNICODE (16 Bits par caractères).</p> <p>Truncate : True / False (Défaut=False) Le fichier est effacé si Truncate est vrai.</p> <p>InsertNL : True / False (Défaut=False) Si InsertNL est vrai, chaque écriture dans le fichier par la commande FILEWRITE générera un saut de ligne.</p> |

Exemple

Cet exemple ouvre le fichier « C:\temp\texte.txt », en écriture, et efface son contenu actuel.

Le programme écrit ensuite du texte dans le fichier.

```

CHGVAR          VAR(&CMD) VALUE('FILEOPEN')
CHGVAR          VAR(&PARM1) VALUE('File="C:\Temp\texte.txt";Truncate=True')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
CHGVAR          VAR(&CMD) VALUE('FILEWRITE')
    
```

```
CHGVAR      VAR(&PARM1) VALUE ('Texte à écrire dans le fichier')
CHGVAR      VAR(&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [FILEWRITE](#)
- [FILECLOSE](#)

Commande *FILEREAD*

Permet de lire le contenu d'un fichier texte (ANSI ou UTF8 sans BOM) ouvert avec la commande FILEOPEN.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('FILEREAD')
CHGVAR      VAR(&PARM1) VALUE('StartBegin=true/false')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>StartBegin :</p> <ul style="list-style-type: none"> - True : commence la lecture au début du fichier. - False : commence la lecture au début de la ligne qui suit la ligne retournée par la dernière commande FILEREAD. <p>La commande FILEREAD retourne chaque ligne terminée par CR LF (Carriage Return Line Feed).</p> <p>Si la dernière ligne a été lue, et qu'on arrive à la fin du fichier, ou bien si le fichier est vide, la commande FILEREAD retourne : %EOF%.</p> |

Exemple

Cet exemple ouvre le fichier « C:\temp\test.txt », et lit son contenu.

Chaque ligne retournée par la commande FILEREAD est contenue dans la variable &RES, et est affichée avec la commande NOP.

On sait qu'on a parcouru tout le fichier dès lors que la commande FILEREAD retourne %EOF%.

```
PGM
DCL          VAR(&RES) TYPE(*CHAR) LEN(2000)
LNCOOPEN
LNCCMD      CMD(FILEOPEN) +
            PARM1('File="c:\temp\test.txt"')
LNCCMDR     CMD(FILEREAD) PARM1('StartBegin=true') +
            RESULT(&RES)
LNCCMD      CMD(NOP) PARM1(&RES)
DOWHILE     COND(&RES *NE '%EOF%')
```

```
LNCCMDR      CMD(FILEREAD) PARM1('StartBegin=false') +  
              RESULT(&RES)  
LNCCMD       CMD(NOP) PARM1(&RES)  
ENDDO
```

```
LNCCMD       CMD(FILECLOSE)  
LNCCLOSE  
ENDPGM
```

Voir aussi

- [FILEWRITE](#)
- [FILECLOSE](#)

Commande *FILEWRITE*

Ecrit dans un fichier texte précédemment ouvert par **FILEOPEN**.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('FILEWRITE')
CHGVAR      VAR(&PARAM1) VALUE('Texte à écrire dans le fichier')
CHGVAR      VAR(&PARAM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Texte à écrire dans le fichier. Le texte est ajouté à la fin du fichier. |

Exemple

Cet exemple ouvre le fichier « C:\temp\texte.txt », en écriture, et efface son contenu actuel.

Le programme écrit ensuite du texte dans le fichier.

```
CHGVAR      VAR(&CMD) VALUE('FILEOPEN')
CHGVAR      VAR(&PARAM1) VALUE('File="C:\Temp\texte.txt";Truncate=True')
CHGVAR      VAR(&PARAM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)

CHGVAR      VAR(&CMD) VALUE('FILEWRITE')
CHGVAR      VAR(&PARAM1) VALUE('Texte à écrire dans le fichier')
CHGVAR      VAR(&PARAM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Remarque

Si la valeur à écrire contient des points virgules, il faut mettre cette valeur entre guillemets :

```
LNCCMD CMD(FILEWRITE) PARM1('"aaa;bbb;ccc;ddd"')
```

Voir aussi

- [FILEOPEN](#)
- [FILECLOSE](#)

Commande FTPCLOSE

Permet de fermer une session FTP ouverte par la commande FTPCONNECT.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('FTPCLOSE')
CHGVAR          VAR(&PARM1) VALUE(' ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Remarque

Ne peut être utilisée que si une session FTP a été préalablement ouverte avec la commande FTPCONNECT.

Exemple

```
LNCCMD CMD(FTPCLOSE)
```

Commande *FTPCONNECT*

Permet d'ouvrir une session FTP avec le serveur FTP spécifié dans les paramètres.

Une fois la connexion ouverte, toutes les autres commandes FTP (FTPPUTFILE...) doivent prendre en compte le répertoire racine du serveur FTP.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('FTPCONNECT')
CHGVAR          VAR(&PARM1) VALUE('
                ServerName="Nom du serveur FTP"
                ;Username="Compte utilisateur FTP"
                ;Password="Mot de passe du compte utilisateur"
                [;SSLEnable=true/false]
                [;ServerPort="Port utilisé"]
                [;Logfile=true/false]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
                &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>ServerName : Nom ou adresse IP du serveur FTP.</p> <p>Username = nom du compte utilisateur autorisé sur le serveur FTP.</p> <p>Password=mot de passe du compte utilisateur.</p> <p>SSLEnable : true/false. Facultatif. Par défaut : false (pas sécurisé). Si SSLEnable=true, le protocole FTP Secure (FTP over TLS/SSL) est utilisé.</p> <p>ServerPort : Port du serveur FTP. Facultatif. Si pas spécifié, le port 21 est utilisé pour les connexions non-sécurisées, et le port 990 pour les connexions sécurisées.</p> <p>Logfile : true/false. Facultatif. Par défaut : false. Si Logfile=true, un fichier logfile est créé dans %TEMP%.</p> |

| | |
|--------|--|
| RESULT | Contient éventuellement le message d'erreur. |
|--------|--|

Exemple

```
LNCCMD CMD(FTPCONNECT)+ PARM1('ServerName="ftp.aura.com";+
username="aura";password="aura"')
```

Commande FTPDELDIR

Permet de supprimer un répertoire complet d'un serveur FTP.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('FTPDELDIR')
CHGVAR          VAR(&PARM1) VALUE('
RemoteDir="Répertoire distant à supprimer"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | RemoteDir : Chemin complet à partir de la racine du serveur FTP et nom du répertoire à supprimer. |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```
LNCCMD          CMD(FTPDELDIR)
PARM1('RemoteDir="/www/Data/temp/Suivi5"')
```

Commande *FTPDELFILE*

Permet de supprimer un fichier d'un serveur FTP.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('FTPDELFILE')
CHGVAR          VAR(&PARM1) VALUE('
RemoteFile="Fichier distant à supprimer"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | RemoteFile : Chemin complet à partir de la racine du serveur FTP et nom du fichier à supprimer. |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```
LNCCMD CMD(FTPDELFILE)
PARM1('RemoteFile="/www/Data/temp/lnc toxls_test5.xlsx"')
```

Commande *FTPGETDIR*

Permet de récupérer tous les fichiers présents dans un répertoire distant sur un serveur FTP.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('FTPGETDIR')
CHGVAR          VAR(&PARM1) VALUE('
LocalDestDir="Répertoire local";
RemoteSrcDir="Répertoire distant"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>LocalDestDir : Chemin complet et nom du répertoire Windows local dans lequel vont être déposés les fichiers distants récupérés.</p> <p>RemoteSrcDir : Chemin complet à partir de la racine du serveur FTP et nom du répertoire contenant les fichiers distants à récupérer.</p> |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```

LNCCMD CMD(FTPGETDIR)
PARM1('RemoteSrcDir="/www/Data/temp/suivi";LocalDestDir="C:\A\suivi"')
    
```

Commande *FTPGETFILE*

Permet de récupérer un fichier présent sur le serveur FTP.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('FTPGETFILE')
CHGVAR          VAR(&PARM1) VALUE('
NewLocalFile="Fichier local récupéré";
RemoteFile="Fichier distant à récupérer"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | NewLocalFile : Chemin complet et nom du fichier local récupéré. RemoteFile : Chemin complet à partir de la racine du serveur FTP et nom du fichier distant à récupérer. |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```
LNCCMD CMD(FTPGETFILE)+
PARM1('NewLocalFile="C:\temp\gettest.xlsx";+
RemoteFile="/www/Data/temp/lnc toxls_test5.xlsx"')
```

Commande FTPLISTDIR

Permet de lister les fichiers d'un répertoire d'un serveur FTP.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('FTPLISTDIR')
CHGVAR          VAR(&PARM1) VALUE('
LocalFile="Fichier local";
RemoteDir="Répertoire distant"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>LocalFile : Chemin complet et nom du fichier local qui contiendra la liste des éléments contenus dans le répertoire distant. Chaque élément du répertoire distant occupera une ligne du fichier local. Un retour à la ligne sépare chaque ligne.</p> <p>RemoteDir : Chemin complet à partir de la racine du serveur FTP et nom du répertoire distant dont on veut lister le contenu.</p> |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```
LNCCMD          CMD(FTPLISTDIR)
PARM1('RemoteDir="/www/Data/temp/mkdir_test";+
LocalFile="c:\temp\mdir_test.csv"')
```

Commande *FTPMKDIR*

Permet de créer un répertoire sur un serveur FTP.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('FTPMKDIR')
CHGVAR          VAR(&PARM1) VALUE('
RemoteDir="Répertoire distant à créer"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | RemoteDir : Chemin complet à partir de la racine du serveur FTP et nom du répertoire à créer. |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```
LNCCMD          CMD(FTPMKDIR) PARM1('/www/Data/temp/Suivi5')
```

Commande *FTPPUTDIR*

Permet de déposer un répertoire complet sur un serveur FTP.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('FTPPUTDIR')
CHGVAR          VAR(&PARM1) VALUE('
LocalSrcDir="Répertoire local"
RemoteDstDir="Répertoire distant"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>LocalSrcDir : Chemin complet et nom du répertoire local contenant les fichiers à déposer.</p> <p>RemoteDstDir : Chemin complet à partir de la racine du serveur FTP et nom du répertoire dans lequel vont être déposés les fichiers.</p> |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```

LNCCMD          CMD(FTPPUTDIR)
PARM1('LocalSrcDir="C:\temp\suivi5";RemoteDstDir="/www/Data/temp+
/suivi5R"')
    
```

Commande *FTPPUTFILE*

Permet de déposer un fichier local sur un serveur FTP.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('FTPPUTFILE')
CHGVAR          VAR(&PARM1) VALUE('
LocalFile="Fichier local à déposer sur le serveur FTP"
NewRemoteFile="Fichier déposé sur le serveur FTP"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | LocalFile : Chemin complet et nom du fichier local à déposer. NewRemoteFile : Chemin complet à partir de la racine du serveur FTP et nom du fichier distant déposé. |
| RESULT | Contient éventuellement le message d'erreur. |

Exemple

```
LNCCMD CMD(FTPPUTFILE)
PARM1('LocalFile="C:\temp\lnctoxls_test5.xlsx";+
NewRemoteFile="/www/Data/temp/lnctoxls_test5.xlsx"')
```

Commande *GETREGE*

Retourne le contenu d'une valeur dans la base de registre.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('GETREGE')
CHGVAR      VAR(&PARM1) VALUE('
Register= "Chemin de la base de registres" ;
Value="Valeur"
')
CHGVAR      VAR(&PARM2) VALUE(' ')
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Register : Chemin de la base de registres.</p> <p>Value : Nom de la valeur.</p> |
| RESULT | Valeur de base de registres en retour, ou *REJ avec message d'erreur. |

Exemple

```

CHGVAR      VAR(&CMD) VALUE('GETREGE')
CHGVAR      VAR(&PARM1) +
            VALUE('Register="HKEY_LOCAL_MACHINE\SOFTWARE+
E\Microsoft\Windows +
NT\CurrentVersion";Value="RegisteredOwner"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RES)
    
```

Commande GETSYSINFO

Retourne des informations sur le système Windows du PC.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('GETSYSINFO')
CHGVAR          VAR(&PARM1) VALUE('ComputerName' | 'UserName' | 'OSVersion
| DefaultPrinter | LNCversion')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Parm1 peut prendre une des valeurs suivantes :</p> <p>ComputerName : En retour, la variable &RESULT contient le nom réseau du PC.</p> <p>UserName : En retour, la variable &RESULT contient le nom de l'utilisateur courant sur le PC.</p> <p>OSVersion : En retour, la variable &RESULT contient la version de Windows.</p> <p>DefaultPrinter : En retour, la variable &RESULT contient le nom de l'imprimante par défaut.</p> <p>LNCVersion : En retour, la variable &RESULT contient la version du serveur Launcher côté Windows.</p> |

Remarque

Si **OSVersion** est utilisé, alors la valeur contenue dans &RESULT sera l'une des suivantes :

| OS | &RESULT |
|------------------------|---------|
| Windows 10 | 10.0 |
| Windows Server 2019 | 10.0 |
| Windows Server 2016 | 10.0 |
| Windows 8.1 | 6.3 |
| Windows Server 2012 R2 | 6.3 |
| Windows 8 | 6.2 |

| | |
|---------------------------|------|
| Windows Server 2012 | 6.2 |
| Windows 7 | 6.1 |
| Windows Server 2008 R2 | 6.1 |
| Windows Server 2008 | 6.0 |
| Windows Vista | 6.0 |
| Windows Server 2003 R2 | 5.2 |
| Windows Server 2003 | 5.2 |
| Windows XP 64-Bit Edition | 5.2 |
| Windows XP | 5.1 |
| Windows 2000 | 5.0 |
| Windows ME | 4.9 |
| Windows 98 | 4.10 |

Exemple

Cet exemple reçoit le nom de l'utilisateur courant, dans la variable &RESULT

```
CHGVAR      VAR(&CMD) VALUE('GETSYSINFO')
CHGVAR      VAR(&PARM1) VALUE('UserName')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Pour connaître la version de Windows installée.

```
CHGVAR      VAR(&CMD) VALUE('GETSYSINFO')
CHGVAR      VAR(&PARM1) VALUE('OSVersion')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

En retour, &RESULT contient la valeur suivante pour un PC Windows 10 :

10.0

Commande *GRAPHSEND*

Cette commande est disponible pour les utilisateurs d'Office 365. L'authentification OAuth (protocole OAuth2) est utilisée pour se connecter au protocole SMTP et envoyer des emails.

Afin de s'authentifier, il faut fournir notamment les éléments suivants :

- Microsoft Entra ID (tenant ID) : paramètre **Tenant**
- ID client de l'application : paramètre **ClientIdVal**
- Valeur secrète de l'application : paramètre **ClientSecretVal**

Bien entendu, ces informations n'apparaîtront pas dans la trace Launcher.

Pour plus d'infos, voici un [lien](#) sur la documentation Microsoft.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE ('GRAPHSEND')
CHGVAR          VAR(&PARM1) VALUE ('
                Server="nom du serveur SMTP"
                ;UserId="compte SMTP qui enverra l'email"
                ;Password="mot de passe du compte"
                ;EmailTo="Destinataires TO"
                ;ClientIdVal="client ID"
                ;ClientSecretVal="client secret"
                ;Tenant="tenant"
                ;Subject="sujet"
                [;Text="Texte de l'email"]
                [;Html="chemin du fichier HTML, corps du mail"]
                [;Attach="liste pièces jointes"]
                [;EmailCc="Destinataires CC"]
                [;EmailBcc="Destinataires BCC"]
                [;Priority=0|1|2]
                [;Importance=0|1|2]
                ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------|---|
| Parm1 ou Parm2 | <p>Server : Nom du serveur SMTP. Généralement : outlook.office365.com.</p> <p>UserId : Compte SMTP utilisé pour l'authentification OAuth et pour l'envoi de l'email (FROM).</p> <p>Password : Mot de passe de UserId.</p> <p>EmailTo : Liste des destinataires TO. Chaque adresse email est séparée par un point-virgule.</p> |

ClientIdVal : Client ID. Utile pour l'authentification.

ClientSecretVal : Client Secret. Utile pour l'authentification.

Tenant : Tenant. Utile pour l'authentification.

Subject : Sujet de l'email.

Text : Texte brut de l'email.

Html : Au lieu d'utiliser le paramètre Text, on peut spécifier ici le chemin complet d'un fichier HTML qui correspondra au corps de l'email.

Attach : Liste des pièces jointes. Il faut spécifier le chemin complet pour chaque pièce jointe. Chaque chemin sera séparé par un point-virgule.

EmailCc : Liste des destinataires CC. Chaque adresse email est séparée par un point-virgule.

EmailBcc : Liste des destinataires BCC. Chaque adresse email est séparée par un point-virgule.

Priority : Priorité de l'email.

0 = non urgent
1 = normal (par défaut)
2 = urgent

Importance : Importance de l'email.

0 = basse
1 = normal (par défaut)
2 = haute

Exemple

L'exemple suivant envoie un email avec un fichier HTML en corps de mail, avec des pièces jointes.

```
LNCCMD CMD (GRAPHSEND) +  
PARM1 ('Server="outlook.office365.com";UserId+  
="test@aura.com";Password="aura"+  
;emailTo="contact@aura.com;+  
tech@aura.com";clientIdVal="ddfgthyp-4444-5555-a+
```



```
fgt-asdf74569b";clientSecretVal="TeV7U~rG+
5555.Y1hPmL9-HG7sC_OPMgbls8156zT";Tenant="4+
745etg9-sese-55e0-5479-5639874df156";Subjec+
t="Test GRAPH";Html="C:\A\ista4.HTML";+
Attach="C:\A\res.pdf;C:\A\F84.pdf;C:\A\image001.jp+
g;c:\A\image002.png"') +
PARM2('Priority=2;Importance=2')
```

Commande IFSPUT

Transfère un fichier du PC vers le système de fichiers intégré (IFS) de l'AS/400.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('IFSPUT')
CHGVAR          VAR(&PARM1) VALUE('Chemin et nom de fichier PC')
CHGVAR          VAR(&PARM2) VALUE('
<Dossier et nom de fichier IFS>
[; Mode=*ALL ]
[; Append ]
[; Text ]
[; RecIO[=nombre] ]
')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin et nom du fichier source PC à transférer vers l'AS/400. |
| Parm2 | <p>Chemin complet vers le fichier ou document de destination dans l'IFS, ou dans le système de fichier de l'AS/400.</p> <p>Si la destination est un fichier dans l'IFS, le chemin doit être exprimé depuis la racine. (/documents/prod_docs/doc01.pdf) Utilisez la commande WRKLNK pour connaître la hiérarchie des répertoires.</p> <p>Si la destination est dans les dossiers partagés, le chemin doit commencer par "/QDLS/". Attention, l'IFS et "Les dossiers partagés" ne représentent pas la même chose. Les dossiers partagés sont dans un sous répertoire de l'IFS : /QDLS.</p> <p>Si la destination est un fichier dans une librairie de l'AS/400, le chemin peut être exprimé sous la forme : /QSYS.LIB/MYLIB.LIB/MYFILE.FILE</p> <p>Il peut aussi être exprimé sous la forme : MYLIB/MYFILE si l'option RecIO est donnée.</p> <p>Le chemin vers le fichier de destination peut être suivi d'options, séparées par un point virgule. Les options sont séparées du chemin vers le fichier, par un point virgule.</p> <p>Mode=*ALL : Donne les droits en lecture et écriture à tous les utilisateurs pour le fichier.</p> <p>Append : Les données seront ajoutées au fichier. Par défaut le</p> |

fichier est effacé pour être remplacé par les nouvelles valeurs.
Si le fichier de destination n'existe pas, il est créé.

Text : Par défaut, le fichier est transféré "tel que", sans conversion.
Si l'option Text est fixée, les données seront converties de texte à texte.

RecIO=Nombre. Cette option indique que le fichier de destination est exprimé sous la forme "LIBRAIRIE/FICHER(MEMBRE) ". Si un nombre est fixée avec l'option **RecIO**, il indique la taille d'enregistrements du fichier si celui-ci doit être créé.
Si le mot **RecIO** seul est fixé, sans nombre affecté, alors le fichier doit exister.

Exemple

L'exemple suivant imprime un document vers un fichier PRN, envoie le fichier PRN vers l'IFS, et le transfère ensuite vers un fichier spool.

```
LNCPRDOC      DOC('C:\Modeles\layout.doc')
               MRGTYPE(*FILE)
               FROMFILE(SP_LAYOUT)
               OUTPUT(*OUTPRN)
               OUTPRN('C:\temp\layout.prn')
               ENDOPT(*APP)

CHGVAR        VAR(&CMD) VALUE('IFSPUT')
CHGVAR        VAR(&PARM1) VALUE('C:\Temp\layout.prn')
CHGVAR        VAR(&PARM2) VALUE('/QDLS/IMAGES/Layout.prn')
CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
               &PARM2 &RESULT)

CRTPRNSPLF    FILE('/QDLS/IMAGES/Layout.prn')
               SPLFNAME(PRNSPLF)
               USRTXT('Document Word')

RMVLNK        OBJLNK('/QDLS/IMAGES/Layout.prn')
CALL          PGM(LNCCLOSE)
```

Voir aussi

- [LNCPRDOC - Commande CL](#)
- [CRTPRNSPLF - Commande CL](#)
- [DBXFER](#)

Commande LISTSCAN

Permet de lister les différents scanners que le serveur Launcher pourra piloter avec la commande [SCAN](#).

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('LISTSCAN')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| RESULT | Le paramètre RESULT contiendra la liste des scanners pilotables par la commande SCAN, séparés par des points-virgules, le tout précédé par "*MSG;". |

Exemple

```
PGM
DCL      VAR(&RESULT) TYPE(*CHAR) LEN(2000)
LNCOPEM

LNCCMDR  CMD(LISTSCAN) RESULT(&RESULT)
LNCCMD   CMD(NOP) PARM1(&RESULT)
LNCCLOSE
ENDPGM
```

Exemple de valeur pour &RESULT : *MSG;HP LJ100 M175 Scan

Commande LNCDBGET

La commande **LNCDBGET** permet de récupérer des données dans une base de données Oracle, MySQL, PostgreSQL, Sybase, DB2 pour iSeries ou Microsoft SQL Server.

Une extension de licence "JDBC" est obligatoire.

Paramètres

La commande LNCDBGET requiert les paramètres suivants :

| | |
|--|-----------------|
| Port du serveur DB | NbPort |
| Type de DB | Type |
| Adresse IPv4 du serveur DB | SvrAddr |
| Utilisateur DB | User |
| Mot de passe utilisateur DB | Password |
| Nom de la base de données | DBName |
| Requête SQL | Query |
| Chemin complet du fichier résultat CSV | CSVFile |

Prérequis

Installation des JARs JDBC sur le PC/serveur hébergeant le serveur Launcher:

Afin de pouvoir accéder aux bases de données Oracle, MySQL, PostgreSQL, Sybase, DB2 pour iSeries ou Microsoft SQL Server, il faut impérativement que les JARs JDBC des bases de données soient présents dans le répertoire d'installation du serveur Launcher.

Par exemple :

*C:\Program Files
(x86)\LAUNCHER400\LAUNCHER_JDBC\LNCClientWrapperJDBC_lib.*

Liste des JARs JDBC officiels à utiliser **obligatoirement** (aucune garantie de fonctionnement pour tout autre JAR utilisé):

- Pour les bases de données Oracle, il faut utiliser le driver Thin fourni par Oracle.
Par exemple pour Oracle Database 10g Release 2, le driver Thin JDBC est nommé **ojdbc14.jar**, et peut être téléchargé en utilisant le lien suivant : <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-10201-088211.html>
- Pour les bases de données MySQL, le driver JDBC **mysql-connector-java-5.1.21-bin.jar** doit être utilisé : <http://www.mysql.com/downloads/connector/j/>



- Pour les bases de données PostgreSQL (\geq version 7.2), le driver JDBC **postgresql-9.2-1000.jdbc4.jar** peut être téléchargé en suivant ce lien : <http://jdbc.postgresql.org/download.html>
- Pour les bases de données Sybase (tous les produits Sybase utilisant JConnect-7_0: Adaptive Server Enterprise, SQL Anywhere, Sybase IQ, et Replication Server), le driver **jconn4.jar** doit être utilisé : <http://www.sybase.fr/products/allproductsa-z/softwaredeveloperkit/jconnect>
- Pour les bases de données Microsoft SQL Server (SQL Server2012, SQL Server 2008 R2, SQL Server 2008, SQL Server 2005, et SQL Azure), le driver **sqljdbc4.jar** doit être utilisé : <http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774>
- Pour les bases de données IBM DB2 pour iSeries, il faut utiliser le driver **jt400.jar** fourni avec IBM System i Access for Windows. Après avoir installé IBM Access sur votre PC, vous pourrez trouver le driver JDBC par exemple ici :
C:\Program Files (x86)\IBM\Client Access\jt400\lib\jt400.jar

Après avoir installé les JARs JDBC, il faut absolument redémarrer le serveur Launcher.

Exemple

Dans cet exemple qui suit, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **PostgreSQL**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 6078
- Utilisateur DB : postgres
- Mot de passe utilisateur DB : aura
- Nom de la database : newold
- Requête SQL : select * from merci where title < "b"

Le fichier de données issues de la base de données est toujours stocké dans :
%TEMP%\CSV_result.csv

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
c:\temp\result.xls

PGM

LNCOPEN

LNCCMD

```
CMD (LNCDBGET) +  
  PARM1 ('Type=postgresql;SvrAddr=192.168.1.7+  
0;NbPort=:6078;User=postgres;Password=aura+  
;DBName=newold;Query=select * from +
```



```
marci where title < +
''b''')+
CSVFile=C:\Users\crobini\AppData\Local\Temp\+
CSV_result.csv')

LNCCMD      CMD (LNCTOXLS) +
            PARM1 ('ToXls=*new;ToSheet=*NONE;ToName=*NON+
            E;ColPref=*NONE;DataSource=C:\Users\crobini+
            \AppData\Local\Temp\CSV_result.csv;RecCnt=32;A+
            utoFmt=*NONE;AutoFit=true;FmtCells=false;Ad+
            dColH=false;ShowDoc=true;SavDoc=C:\temp\res+
            ult.xls;SavFmt=*XLS')

LNCCLOSE
ENDPGM
```

Commande MAILATT

Compatible SMTP, MAPI et Lotus Notes

Ajout d'un fichier attaché au message actuellement en cours d'élaboration.

Vous devez avoir fait appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Il existe deux prototypes différents suivant l'interface que vous utilisez.

Syntaxe

1) Première syntaxe (SMTP, MAPI et Lotus Notes)

```
CHGVAR          VAR(&CMD) VALUE('MAILATT')
CHGVAR          VAR(&PARM1) VALUE('Fichier')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
```

| Paramètres | |
|------------|---|
| Parm1 | Chemin complet vers le fichier à attacher |

2) Deuxième syntaxe (SMTP)

Cette commande peut aussi permettre d'embarquer des images dans un message au format HTML.

Le fichier HTML doit être ajouté comme corps du message avec la commande **MAILBODYF** avec l'option HTML.

Il faut que le tag insérant l'image dans la page HTML ait la propriété 'src' avec une valeur du type "src=cid:IMAGE001.GIF" :

Il faut également que le deuxième paramètre de la commande MAILATT ait une valeur du genre HEADER="CONTENT-ID: <IMAGE001.GIF>.

Attention tous les lecteurs d'e-mail ne gèrent pas cette fonctionnalité !

```
CHGVAR          VAR(&CMD) VALUE('MAILATT')
CHGVAR          VAR(&PARM1) VALUE('Fichier')
CHGVAR          VAR(&PARM2)
```



```
CALL VALUE ('HEADER="CONTENT-ID:<IMAGE001.GIF>')
PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

| Paramètres | |
|------------|--|
| Parm1 | Chemin complet vers le fichier à attacher |
| Parm2 | Paramètre optionnel. Il permet d'embarquer des images correspondants à un message au format HTML . La syntaxe est : HEADER="CONTENT-ID:<identifiant>" |

Exemples

```
LNCCMD CMD(MAILATT) +
PARM1 ('\\ADX01\SUPPORT\TEST\LOGOA2.JP+
G') +
PARM2 ('HEADER="CONTENT-ID:<IMAGE001.GIF>')
```

```
LNCCMD CMD(MAILATT) +
PARM1 ('\\ADX01\SUPPORT\TEST\IMAGE001.+
GIF')
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILTO](#), ajoute des destinataires au message
- [MAILPRTY](#), définition d'une priorité du message
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILREPORT](#), pour activer le suivi du message
- [MAILSEND](#), envoie le message
- [MAILEND](#), libère toutes les structures allouées par le message
- [MAILBODYF](#) définir un corps de texte au format HTML
- [MAILCC](#) définir les destinataires secondaires du message
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP

Commande MAILBODYF

Compatible SMTP

Envoi un fichier HTML ou Texte comme corps du message.

Vous devez d'abord avoir fait appel à la commande [MAILPREP](#) pour commencer l'élaboration du message.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('MAILBODYF')
CHGVAR          VAR(&PARM1) VALUE('<chemin fichier HTML ou texte')
CHGVAR          VAR(&PARM2) VALUE('HTML|TXT')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin complet vers le fichier HTML à insérer. |
| Parm2 | Nom du format utilisé : HTML ou TXT. |

Exemple

```
LNCCMD          CMD(MAILBODYF) +
                PARM1 ('\\ADX01\SUPPORT\XYLEM\TEST\TEST2.HTM+
L') PARM2 ('HTML')
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILSEND](#), envoie le message

- [MAILEND](#), libère toutes les structures allouées par le message.
- [MAILCC](#), pour définir les destinataires secondaires du message.
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP



Commande MAILCC

Compatible SMTP.

Définition des destinataires secondaires cachés ou non.

Vous devez avoir fait appel à la commande [MAILPREP](#) pour commencer l'élaboration du message.

S'il y a plusieurs destinataires secondaires cachés ou non, il faut qu'ils soient séparés par des points virgules.

Tous les formats d'adresse peuvent être utilisés et combinés (tech<tech@easycom-aura.com> ou tech@easycom-aura.com).

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('MAILCC')
CHGVAR          VAR(&PARM1) VALUE('tech<tech@easycom-aura.com>')
CHGVAR          VAR(&PARM2) VALUE('sales@Easycom-aura.com;info@Easycom-aura.com')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Liste des destinataires secondaires non cachés séparés par des points virgules. |
| Parm2 | Liste des destinataires secondaires cachés séparés par des points virgules. |

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILSEND](#), envoie le message
- [MAILEND](#), libère toutes les structures allouées par le message
- [MAILBODYF](#), pour définir un corps de texte au format HTML
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP





Commande **MAILEND**

Compatible SMTP, MAPI et Lotus Notes

Libération des structures utilisées pour l'élaboration du message.

Vous devez faire appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('MAILEND')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
```

Exemple

```
LNCCMD          CMD (MAILEND)
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILTO](#), ajoute des destinataires au message
- [MAILPRTY](#), définition d'une priorité du message
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILREPORT](#), pour activer le suivi du message
- [MAILSEND](#), envoie le message
- [MAILBODYF](#) définir un corps de texte au format HTML
- [MAILCC](#) définir les destinataires secondaires du message
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP

Commande MAILLATT

Compatible Lotus Notes

Liste les pièces jointes au dernier message listé par MAILMSG.

Cette commande retourne dans le paramètre &RESULT les informations demandées sur le premier message ou le prochain message d'un dossier Lotus. Si la fin de la liste des messages est atteinte, la commande retourne FALSE dans &RESULT.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('MAILLATT')
CHGVAR      VAR(&PARM1) VALUE('
              [Start=True/False;]
              ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                              &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | Start : Si Start=true, l'appel de la commande retournera le premier fichier attaché au message. |
| RESULT | Après l'appel de la commande, le paramètre &RESULT contient : FALSE si la fin de la liste est atteinte Type ="Type de fichier"; Name ="Nom du fichier attaché" ; Path ="Chemin vers le fichier attaché" ; |

Exemple

```
CHGVAR VAR(&CMD) VALUE('MAILLATT')
CHGVAR VAR(&PARM1) VALUE('Start=True')
CHGVAR VAR(&RESULT) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2 &RESULT)

IF COND(&RESULT *EQ 'FALSE') THEN(GOTO CMDLBL(END))
```

```
CHGVAR VAR(&NAME) VALUE(' ')
CALL PGM(LNCGETKW) PARM(&HANDLE &RES 'Name ' &NAME)
```

Commande MAILMSG

Compatible Lotus Notes

Liste les messages de la boîte de réception.

Cette commande retourne dans le paramètre &RESULT les informations demandées sur le premier message ou le prochain message d'un dossier Lotus. Si la fin de la liste des messages est atteinte, la commande retourne FALSE dans &RESULT.

Syntaxe

```
CHGVAR VAR(&CMD) VALUE('MAILMSG')
CHGVAR VAR(&PARM1) VALUE('
Folder=Dossier à lister";
Start=True/False;
Subject=True/False;
SenderName=True/False;
UnReadOnly=True/false;
')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>Folder : Indique le dossier dont on veut lister le contenu. Si "Folder" n'est pas indiqué, le dossier de réception sera listé.</p> <p>Start : Si "Start » est vrai, l'appel de la commande retournera le premier message du dossier. Si "Start" est faux, le message suivant le précédent appel est retourné.</p> <p>Subject : Si "Subject" est vrai, le texte du sujet est retourné dans le paramètre &RESULT, sous la forme : Subject="Sujet du message" ;</p> <p>SenderName : Si "SenderName" est vrai, le nom de l'expéditeur du message est retourné dans le paramètre &RESULT, sous la forme :</p> |

SenderName="Nom de l'expéditeur" ;

UnReadOnly : "UnReadOnly" est vrai par défaut. Seuls les messages non lus sont listés.

RESULT Après l'appel de la commande, le paramètre &RESULT contient :
FALSE si la fin de la liste est atteinte,
Subject="Sujet du message"; si l'option "Subject=True" était donnée.
SenderName="Nom de l'expéditeur", si l'option "SenderName=True" était donnée.

Utilisez le programme [LNCGETKW](#) pour analyser le contenu de &RESULT.

Exemple

```
CHGVAR VAR(&CMD) VALUE('MAILMSG')
CHGVAR VAR(&PARM1) VALUE('Subject=True;SenderName=True; +
folder="domino/aura!!mail/box.nsf"; +
UnReadOnly=false')

CHGVAR VAR(&PARM2) VALUE('Subject=True;SenderName=True')

CHGVAR VAR(&RESULT) VALUE('*')

CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2 &RESULT)

IF COND(&RESULT *EQ 'FALSE') THEN(GOTO CMDLBL(END))

CHGVAR VAR(&SUBJ) VALUE(' ')
CALL PGM(LNCGETKW) PARM(&HANDLE &RESULT 'Subject ' &SUBJ)
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [LNCGETKW - Programme API](#)
- [MAILLATT](#)
- [MAILMSG](#)
- [MAILMCHG](#)
- [MAILMDLT](#)
- [MAIXATT](#)





Commande MAILMCHG

Compatible Lotus Notes

Change des propriétés du dernier message listé par la commande MAILMSG.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('MAILMCHG')
CHGVAR      VAR(&PARM1) VALUE('
              [AsRead=True/False]
              ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
              &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | AsRead : Si AsRead est vrai, le message est marqué comme "lu". Le message ne peut pas être marqué comme "non lu". |

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILLATT](#)
- [MAILMSG](#)
- [MAILMCHG](#)
- [MAILMDLT](#)
- [MAIXATT](#)

Commande MAILMDLT

Compatible Lotus Notes

Supprime le dernier message listé par la commande MAILMSG.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('MAILMDLT')
CHGVAR      VAR(&PARM1) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILLATT](#)
- [MAILMSG](#)
- [MAILMCHG](#)
- [MAILMDLT](#)
- [MAIXATT](#)

Commande MAILPREP

Compatible SMTP, MAPI et Lotus Notes.

Préparation d'un message destiné à être envoyé par le système de messagerie.

Pour tout message à envoyer, débutez par l'appel à cette commande.
Il existe deux prototypes différents pour cette fonction selon l'interface que vous utilisez.

Syntaxe

1) Première syntaxe (MAPI et Lotus Notes)

```
CHGVAR          VAR(&CMD) VALUE('MAILPREP')
CHGVAR          VAR(&PARM1) VALUE('Destinataires')
CHGVAR          VAR(&PARM2) VALUE('Sujet')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                               &PARM2 &RESULT)
```

| Paramètres | |
|------------|--|
| Parm1 | Liste des destinataires séparés par des points virgules . |
| Parm2 | Sujet du message |

2) Deuxième syntaxe (SMTP)

Attention : En SMTP, exécutez la commande MAILSMTP avant la commande MAILPREP.

```
CHGVAR          VAR(&CMD) VALUE('MAILPREP')
CHGVAR          VAR(&PARM1) VALUE('Destinataires')
CHGVAR          VAR(&PARM2) VALUE('
[subject="sujet";]
[priority=0 à 5;]
[charset=type encodage;]
[report]
')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                               &PARM2 &RESULT)
```

| Paramètres |
|------------|
|------------|



| | |
|-------|--|
| Parm1 | Liste des destinataires déparés par des points virgules . |
| Parm2 | Chaîne de caractères contenant différentes propriétés*. Toutes les propriétés sont optionnelles. |

| *Propriétés | |
|-----------------|---|
| subject | Sujet du message. |
| priority | Niveau d'importance du message (de 0 à 5, du moins au plus important). |
| charset | Définit le type d'encodage des caractères utilisé. Valeurs possibles : ASCII, UTF-8, UTF-7, UTF-32, Unicode. |
| report | Si cette propriété est présente dans le deuxième paramètre, alors la demande d'accusé de réception est activée. |

Exemple

```
LNCCMD CMD (MAILPREP) PARM1 ('tech@easycom-aura.com')  
PARM2 ('subject="test";priority=5;report')
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILTO](#), ajoute des destinataires au message
- [MAILPRTY](#), définition d'une priorité du message
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILREPORT](#), pour activer le suivi du message
- [MAILSEND](#), envoie le message
- [MAILEND](#), libère toutes les structures allouées par le message
- [MAILBODYF](#) définir un corps de texte au format HTML
- [MAILCC](#) définir les destinataires secondaires du message
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP



Commande MAILPTY

Compatible MAPI et Lotus Notes.

Définit la priorité du message en cours de préparation.

Vous devez d'abord avoir fait appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Les différents niveaux de priorité sont :

HIGH : haute priorité,

NORMAL : priorité normale (par défaut),

LOW : basse priorité.

Pour SMTP, la priorité du message se fixe avec la commande MAILPREP.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('MAILPTY')
CHGVAR          VAR (&PARM1) VALUE (' [*HIGH | *NORMAL | *LOW] ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---------------------------------------|
| Parm1 | *HIGH OU *NORMAL (par défaut) OU *LOW |

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILTO](#), ajoute des destinataires au message
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILREPORT](#), pour activer le suivi du message
- [MAILSEND](#), envoie le message
- [MAILEND](#), libère toutes les structures allouées par le message.

Commande **MAILREPORT**

Compatible MAPI et Lotus Notes

Active la demande d'avis de réception du message.

Vous devez faire appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Pour **SMTP**, la demande d'avis de réception est gérée par la commande [MAILPREP](#).

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('MAILREPORT')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILTO](#), ajoute des destinataires au message
- [MAILPRTY](#), définition d'une priorité du message
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILSEND](#), envoie le message
- [MAILEND](#), libère toutes les structures allouées par le message

Commande MAILSEND

Compatible SMTP, MAPI et Lotus Notes

Envoi du message actuellement en cours d'élaboration.

Vous devez faire appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('MAILSEND')
CHGVAR      VAR(&PARM1) VALUE('
[ Retry=Nombre d'essais]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Retry : Nombre d'essais pour l'envoi de l'email en cas d'échec. |

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILTO](#), ajoute des destinataires au message
- [MAILPRTY](#), définition d'une priorité du message
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILREPORT](#), pour activer le suivi du message
- [MAILEND](#), libère toutes les structures allouées par le message
- [MAILBODYF](#) définir un corps de texte au format HTML
- [MAILCC](#) définir les destinataires secondaires du message
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP



Commande MAILSMTP

Compatible SMTP.

Prépare la connexion avec le serveur SMTP.

Pour tout envoi de messages par le protocole SMTP, il faut débiter le programme par l'appel de cette commande.

Le premier paramètre de la commande reçoit la liste des propriétés séparées entre elles par des ";".

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('MAILSMTP')
CHGVAR          VAR(&PARM1) VALUE('
Server="serveur SMTP";
Sender="adresse email de l'expéditeur du mail"
[ ;UserId="Profil utilisateur smtp"]
[ ;password=Mot de passe profil utilisateur smtp]
[ ;reply="adresse email recevant les réponses"]
[ ;logfile="chemin du fichier logfile"]
[ ;port=numéro du port du serveur SMTP ]
[ ;enableSSL=true/false ]
[ ;SSLType=SMTPS/TLS ]
[ ;Sign=true/false ]
[ ;Encrypt=true/false ]
[ ;Certif="chemin du certificat .p12"]
[ ;PwdCertif=Mot de passe du certificat]
[ ;Timeout=en secondes]
[ ;SecureHeaders=true/false ]
[ ;WrapperSubject="Sujet du wrapper RFC 822"]
[ ;checkMail= true/false]
[ ;useTestMode= true/false]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Liste des propriétés* de la connexions SMTP, séparées entre elles par des ";". |

| *Propriétés | |
|---------------|--|
| sender | Adresse de la personne qui envoie le message. Obligatoire. |

| | |
|------------------|---|
| Server | Adresse IP ou nom DNS du serveur SMTP. Obligatoire. |
| UserId | Profil de l'utilisateur pour la connexion SMTP. Optionnel. |
| password | Mot de passe de l'utilisateur. Optionnel. |
| reply | Adresse de la personne qui réceptionne les réponses au message envoyé, si elle est différente de l'expéditeur. Optionnel. |
| logfile | Chemin et nom du fichier où seront inscrites les traces des envois effectués. Permet de s'assurer que l'envoi du message s'est déroulé sans erreur. Ce fichier est formaté par des séparateur(;), pour pouvoir être importé dans une base de données ou Excel. Optionnel. |
| port | Optionnel. Port du serveur SMTP dans le cas où le serveur n'utilise pas le port 25. Pour le protocole TLS, généralement le port 587 est utilisé. Et pour le protocole SMTPS (SMTP over SSL), c'est généralement le port 465. |
| EnableSSL | True/False (par défaut). Si true, Launcher utilise par défaut le protocole TLS. Vous pouvez aussi le protocole SMTPS (SMTP over SSL) en utilisant le paramètre SSLType . |
| SSL Type | Deux valeurs possibles : TLS (par défaut) et SMTPS . TLS : Launcher prend en charge l'Extension de Service SMTP pour SMTP sécurisé sur le protocole TLS comme défini dans RFC 3207. Dans ce mode, la session SMTP commence sur un canal non chiffré, puis une commande STARTTLS est émise par le client au serveur pour basculer vers une communication sécurisée via SSL. Consultez RFC 3207 publiée par l'IETF (Internet Engineering Task Force) pour plus d'informations. Généralement le port 587 est utilisé. SMTPS : Appelé aussi SMTP/SSL ou SMTP over SSL. Le port 465 est généralement utilisé. La communication est sécurisée via SSL dès le début de la connexion. |
| Sign | True/False (par défaut). Si true, Launcher signe l'email envoyé, |

| | |
|-----------------------|--|
| | <p>comme défini dans RFC 8551. Il utilise le certificat (.p12) spécifié avec le paramètre Certif.</p> <p>Conforme à S/MIME Version 4.0 et RFC 8551. Consultez RFC 8551 publiée par l'IETF (Internet Engineering Task Force) pour plus d'informations.</p> |
| Encrypt | <p>True/False (par défaut). Si true, Launcher va chiffrer le corps de l'email envoyé comme défini dans RFC 8551. Il utilise le certificat (.p12) spécifié avec le paramètre Certif.</p> <p>Conforme à S/MIME Version 4.0 et RFC 8551. Consultez RFC 8551 publiée par l'IETF (Internet Engineering Task Force) pour plus d'informations.</p> |
| Certif | <p>Chemin complet du certificat (.p12) utilisé pour chiffrer et/ou signer l'email.</p> |
| PwdCertif | <p>Mot de passe du certificat (.p12) utilisé pour chiffrer et/ou signer l'email.</p> |
| Timeout | <p>Définit le délai d'expiration de l'opération en secondes.</p> |
| SecureHeaders | <p>Protection des en-têtes grâce à l'utilisation d'un message RFC 822. Un wrapper message/rfc822 est utilisé pour appliquer les services de sécurité S/MIME aux headers.</p> <p>Voir norme RFC 8551 pour plus de détails. L'email sera reçu en pièce jointe par le destinataire.</p> |
| WrapperSubject | <p>Si SecureHeaders=true, par défaut le destinataire recevra un email sans objet et une pièce jointe contenant l'email spécifié. Vous pouvez spécifier ici un sujet afin d'éviter que l'email aille dans les spams.</p> |
| checkMail | <p>True/False (par défaut). Si true, une vérification des adresses email des destinataires sera effectuée selon la norme RFC 5322.</p> <p>La vérification sera effective pour les commandes MAILPREP, MAILTO et MAILCC.</p> |
| useTestMode | <p>True/False (par défaut). Si true, un serveur SMTP de test, qui utilise un certificat auto signé, peut être utilisé. A ne pas utiliser en production.</p> |

Exemple

```
LNCCMD      CMD (MAILSMTP) +  
PARM1 ('SERVER="SMTP.AURA.FR";SENDER="TECH+  
@EASYCOM-AURA.COM";REPLY="TECH@EASYCOM-+  
AURA.COM";logfile="c:\temp\logaurasmtplib")
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message ;
- [MAILATT](#), pour ajouter un fichier attaché au message ;
- [MAILTEXT](#), pour définir un corps de texte ;
- [MAILSUBJ](#), ajoutant un sujet au message ;
- [MAILSEND](#), envoie le message ;
- [MAILEND](#), libère toutes les structures allouées par le message.
- [MAILBODYF](#) définir un corps de texte au format HTML.
- [MAILCC](#) définir les destinataires secondaires du message.

Commande *MAILSUBJ*

Compatible SMTP, MAPI et Lotus Notes

Ajout d'un sujet au message actuellement en cours d'élaboration.

Vous devez faire appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('MAILSUBJ')
CHGVAR          VAR(&PARM1) VALUE('Sujet')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|------------------|
| Parm1 | Sujet du message |

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTEXT](#), pour définir un corps de texte
- [MAILTO](#), ajoute des destinataires au message
- [MAILPRTY](#), définition d'une priorité du message
- [MAILREPORT](#), pour activer le suivi du message
- [MAILSEND](#), envoie le message
- [MAILEND](#), libère toutes les structures allouées par le message
- [MAILBODYF](#) définir un corps de texte au format HTML
- [MAILCC](#) définir les destinataires secondaires du message
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP

Commande *MAILTEXT*

Compatible SMTP, MAPI et Lotus Notes

Ajout d'un texte au message actuellement en cours d'élaboration.

Vous devez faire appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('MAILTEXT')
CHGVAR          VAR (&PARM1) VALUE ('Texte')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|------------------|
| Parm1 | Corps du message |

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message
- [MAILATT](#), pour ajouter un fichier attaché au message
- [MAILTO](#), ajoute des destinataires au message
- [MAILPRTY](#), définition d'une priorité du message
- [MAILSUBJ](#), ajoutant un sujet au message
- [MAILREPORT](#), pour activer le suivi du message
- [MAILSEND](#), envoie le message
- [MAILEND](#), libère toutes les structures allouées par le message
- [MAILBODYF](#) définir un corps de texte au format HTML
- [MAILCC](#) définir les destinataires secondaires du message
- [MAILSMTP](#) définit les propriétés pour se connecter au serveur SMTP

Commande MAILTO

Compatible SMTP, MAPI et Lotus Notes

Ajout de destinataires du message actuellement en cours d'élaboration.

Vous pouvez spécifier le type d'envoi qui leur sera fait (TO, CC, BCC).

Vous devez faire appel à la commande [MAILPREP](#) pour commencer l'élaboration d'un message.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('MAILTO')
CHGVAR          VAR(&PARM1) VALUE('Destinataires')
CHGVAR          VAR(&PARM2) VALUE('[*TO|*CC|*BCC]')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Liste de destinataires séparés par des points virgules |
| Parm2 | Type d'envoi fait aux destinataires : *TO : valeur par défaut *CC : copie *BCC : copie cachée |

Exemple

```
LNCCMD CMD(MAILTO) PARM1('tech@easycom-aura.com';'info@easycom-aura.com') PARM2('*TO')
```

```
LNCCMD CMD(MAILTO) PARM1('tech@easycom-aura.com';'info@easycom-aura.com') PARM2('*CC')
```

```
LNCCMD CMD(MAILTO) PARM1('tech@easycom-aura.com';'info@easycom-aura.com') PARM2('*BCC')
```

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILPREP](#), pour préparer le message



- **MAILATT**, pour ajouter un fichier attaché au message
- **MAILTEXT**, pour définir un corps de texte
- **MAILPRTY**, définition d'une priorité du message
- **MAILSUBJ**, ajoutant un sujet au message
- **MAILREPORT**, pour activer le suivi du message
- **MAILSEND**, envoie le message
- **MAILEND**, libère toutes les structures allouées par le message

Commande MAIXATT

Compatible Lotus Notes.

Extrait une pièce jointe du dernier message listé par MAILLATT.

Cette commande retourne dans le paramètre &RESULT les informations sur la pièce extraite du message.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('MAIXATT')
CHGVAR          VAR(&PARM1) VALUE('
File="Chemin et nom du fichier destination";
[Note=True/False]
')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | File : Chemin vers le fichier de destination qui contiendra l'objet extrait. Note : Si Note est vrai, la commande extrait le texte du message. |
| RESULT | Après l'appel de la commande, le paramètre &RESULT contient : FALSE si aucun fichier n'est extrait, Type="Type de fichier"; Name="Nom du fichier extrait" ; Path="Chemin initial vers le fichier" ; |

Voir aussi

- [LNCSNDMAIL - Commande CL](#)
- [MAILLATT](#)
- [MAILMSG](#)
- [MAILMCHG](#)
- [MAILMDLT](#)
- [MAIXATT](#)





Commande MENU

Construit une barre de menu personnalisée indépendante de toute application.

MENU doit être utilisé avec la commandes **MENUWAIT**.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('MENU')
CHGVAR      VAR(&PARM1) VALUE('Captions="Libellés des boutons ;..."
                        [;Tips="Bulle d'aide ;..."]
                        [;MenuName="Nom du menu"]')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Captions= Permet de donner le libellé de chaque bouton de la nouvelle barre de menu. Les libellés sont séparés entre eux par un point virgule (;). Le nombre de libellés détermine le nombre de boutons de la barre de menu. Tips= Permet d'afficher une bulle d'aide pour chaque bouton. MenuName= Permet de donner un nom au menu. |

Remarque

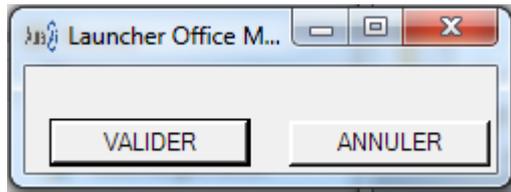
Lors de l'appel de la commande MENUWAIT, le programme AS/400 se met en attente d'une action de la part de l'utilisateur. Il reprend la main sur l'action d'un des boutons du nouveau menu, ou quand le menu est fermé.

Au retour de la commande MENUWAIT, la variable &RESULT contient le numéro sur 5 chiffres du bouton actionné (De 1 à n).

Le menu se ferme automatiquement lorsqu'une action a eu lieu.

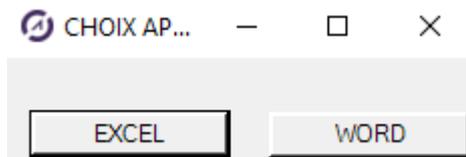
Exemple

```
CHGVAR      VAR(&CMD) VALUE('MENU')
CHGVAR      VAR(&PARM1) +
            VALUE('CAPTIONS="VALIDER;ANNULER";TIPS="VAL+
            IDER LE DOCUMENT;ANNULER"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT)
```



```

CHGVAR      VAR (&CMD) VALUE ('MENU')
CHGVAR      VAR (&PARM1) +
            VALUE ('CAPTIONS="EXCEL;WORD";TIPS="OUVRIR +
            EXCEL;OUVRIR WORD";MENUNAME="CHOIX APPLI +
            OFFICE"')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
            &PARM2 &RESULT)
    
```



Commande MENUWAIT

Attend un évènement sur le menu créé par la commande MENU.

Le programme client attend une action sur le menu personnalisé, ou bien la fermeture du menu, pour continuer son exécution.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('MENUWAIT')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Exemple

```
CHGVAR      VAR(&CMD) VALUE('MENU')
CHGVAR      VAR(&PARM1) +
            VALUE('CAPTIONS="VALIDER;ANNULER";TIPS="VAL+
            IDER LE DOCUMENT;ANNULER"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/* ATTENTE CLICK SUR BOUTON */
CHGVAR      VAR(&CMD) VALUE('MENUWAIT')
CHGVAR      VAR(&OPT) VALUE(' ')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
IF          COND(&RESULT *NE '00000') THEN(GOTO +
            CMDLBL(WORDCLICK))

GOTO        CMDLBL(FINISH)

WORDCLICK:
SNDSMSG     MSG('CLICK = ' *TCAT &RESULT) TOUSR(QPGMR)

FINISH:
```

Commande MIN

Permet la réduction de la fenêtre courante.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('MIN')
CHGVAR          VAR(&PARM1) VALUE('[Numéro de fenêtre]')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | " <i>Numéro de fenêtre</i> " est facultatif. Ce numéro a été retourné par un précédent appel à la commande STO . S'il est spécifié, c'est la fenêtre dont le numéro est donné qui sera minimisée. S'il est absent, la fenêtre active courante sera minimisée. |
| RESULT | En sortie, ce paramètre contient le numéro de la fenêtre qui a été minimisée, sur 10 chiffres. Si un numéro de fenêtre n'a pas été donné dans Parm1 , c'est le numéro de la fenêtre active qui est retourné, et cette fenêtre pourra être restaurée par la commande RCL . |

Voir aussi

- [RCL](#)
- [STO](#)

Commande *MOVEFILE*

Permet de déplacer un fichier d'un répertoire vers un autre sur le PC.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('MOVEFILE')
CHGVAR          VAR(&PARM1) VALUE('
File="Fichier à déplacer"
;NewFile="Fichier déplacé"
[;Replace=true/false]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | File : Chemin complet et nom du fichier à déplacer. NewFile : Chemin complet et nom du fichier déplacé. Replace = true, pour remplacer automatiquement le fichier remplacé s'il existe déjà. Par défaut, Replace=false. |
| RESULT | Contient éventuellement le message d'erreur. |

Remarque

Si le fichier destination existe déjà et que Replace=false, alors une erreur sera retournée. Exemple dans la trace:

```
6540 2019-4-17 10:13:4.808 ERROR: the destination file already
exists, and Replace=false
6540 2019-4-17 10:13:4.811 *REJ
```

Exemple

```
LNCCMD CMD(MOVEFILE) PARM1('File="C:\temp\SBOOK.csv"; +
NewFile="C:\temp\TEST\SBOOK.csv";replace=true')
```

Commande MSGBOX

Affiche une boîte à message sur le poste Windows.

A ne pas utiliser avec Launcher en mode service Windows.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('MSGBOX')
CHGVAR      VAR(&PARM1) VALUE('
Text="Texte du message";
Caption="Titre de la fenêtre"
')
CHGVAR      VAR(&PARM2) VALUE('
Type=Type du bouton
[;Default=Numéro du bouton par défaut]
')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

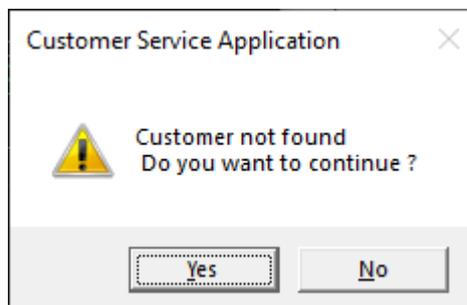
Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Parm1 contient le titre et le texte à afficher dans la boîte à message :</p> <p>Text=Texte dans la boîte à message.</p> <p>Caption=Titre de la fenêtre.</p> |
| Parm2 | <p>Parm2 contient les options de la boîte à message :</p> <p>Type=type de la boîte à message. Il peut avoir l'une de ces valeurs suivantes : MB_ABORTRETRYIGNORE, MB_OK, MB_OKCANCEL, MB_RETRYCANCEL, MB_YESNO, MB_YESNOCANCEL.</p> <p>L'une des valeurs suivantes peut être ajoutée (avec le signe +) pour afficher une icône dans la boîte à message : MB_ICONEXCLAMATION, MB_ICONWARNING, MB_ICONINFORMATION, MB_ICONASTERISK, MB_ICONQUESTION, MB_ICONSTOP, MB_ICONERROR, MB_ICONHAND</p> <p>Default= Numéro du bouton à rendre actif par défaut. Optionnel.</p> |
| RESULT | <p>Contient le numéro du bouton sur lequel l'utilisateur a cliqué.</p> |

Exemple

Cet exemple montre l'affichage d'une boîte à message, avec le bouton "Oui" et "Non", et l'icône avec le point d'exclamation.

```
CHGVAR          VAR(&CMD) VALUE('MSGBOX')
CHGVAR          VAR(&PARM1) VALUE('Text="Customer not found %LF%
Do you want to continue ?";Caption="Customer Service
Application"')
CHGVAR          VAR(&PARM2) VALUE('Type= MB_ICONEXCLAMATION+
MB_YESNO')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```



Remarque

%LF% : Permet d'insérer un saut de paragraphe dans le texte à afficher.

Commande NCL

Permet la création d'une nouvelle colonne dans le fichier source créé par la commande CRF.

La colonne sera insérée à droite des colonnes déjà existantes.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('NCL')
CHGVAR      VAR(&PARM1) VALUE('NomColonne')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---------------------------|
| Parm1 | Nom de la colonne à créer |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('NCL')
CHGVAR      VAR(&PARM1) VALUE('Colonne1')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [CRF](#)
- [ENF](#)
- [NLN](#)
- [VAL](#)

Commande NLN

Permet la création d'une nouvelle ligne dans le fichier source créé par la commande CRF.

La ligne sera ajoutée à la fin du fichier source.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('NLN')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [CRF](#)
- [ENF](#)
- [NCL](#)
- [VAL](#)

Commande NOP

Permet de mettre un commentaire dans l'onglet Trace de l'administration LAUNCHER.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('NOP')
CHGVAR          VAR(&PARM1) VALUE('Commentaire ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|-------------|
| Parm1 | Commentaire |

Exemple

```
CHGVAR          VAR(&CMD) VALUE('NOP')
CHGVAR          VAR(&PARM1) VALUE('Ceci est un commentaire')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande *OLECALL*

Permet d'accéder à une méthode ou une propriété d'une instance d'un objet précédemment créée par la commande *OLECREATE*.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('OLECALL')
CHGVAR          VAR(&PARM1) VALUE('"Identifiant de l'instance"')
CHGVAR          VAR(&PARM2) VALUE('Méthode ou propriété à exécuter')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Identifiant de l'instance de l'objet. Cet identifiant a été retourné par un précédent appel à <i>OLECREATE</i> . |
| Parm2 | Texte de la méthode ou de la propriété à exécuter. Exemples : ActiveDocument.SaveAs("C:\MyDoc\Doc1.doc") Sauve le document courant. ActiveDocument.Words.Count Retourne le nombre de mots du document. |
| RESULT | En sortie, le paramètre &RESULT contient une valeur éventuellement retournée par la méthode appelée, ou la valeur d'une propriété demandée. |

N'utilisez cette commande que si vous connaissez bien la programmation OLE sous Windows, et si vous connaissez bien l'objet avec lequel vous devez communiquer.

Exemple

Voir la commande [OLECREATE](#).

Voir aussi

- [OLECREATE](#)

Commande **OLECREATE**

Permet de créer une instance d'un objet OLE.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('OLECREATE')
CHGVAR      VAR(&PARM1) VALUE('"Référence vers l'objet"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Référence vers l'application dont on veut créer une instance. Par exemple, pour créer une instance de Microsoft Access, on indiquera : "Access.application". Pour créer une instance de Lotus : "Notes.NotesSession". Les références aux objets sont dans la base de registre de Windows, sous la clé HKEY_CLASSES_ROOT. |
| RESULT | En sortie, le paramètre &RESULT contient l'identifiant de l'instance de l'objet créé, sur 10 caractères. Cet identifiant sera utilisé par la commande OLECALL . |

N'utilisez cette commande que si vous connaissez bien la programmation OLE sous Windows, et si vous connaissez bien l'objet avec lequel vous devez communiquer.

Exemple

```
LNCCMDR     CMD(OLECREATE) PARM1 (ACCESS.APPLICATION) +
             RESULT (&RES)

LNCCMD      CMD(OLECALL) PARM1 (&RES) +
             PARM2 ('NewCurrentDatabase("LAUNCH24.MDB")')
```

```
LNCCMD      CMD(OLECALL) PARM1 (&RES) +
             PARM2 ('DoCmd.TransferText(0;"SP_CUST34";"C+
:\AuraData\ACCESS_2010_test\LNC003_virgule.+
TXT";true;;1200)')
```

```
LNCCMD      CMD(OLERELEASE) PARM1 (&RES)
```

Voir aussi

- [OLECALL](#)

Commande *OLERELEASE*

Permet de libérer un objet créé par la commande *OLECREATE*.

La commande **LNCCLOSE** permet de tout libérer automatiquement.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('OLERELEASE')
CHGVAR          VAR(&PARM1) VALUE('"Référence vers l'objet"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                             &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Référence vers l'application dont on veut libérer l'objet. |

Exemple

Voir la commande [OLECREATE](#).

Voir aussi

- [OLECALL](#)
- [OLECREATE](#)

Commande PARSEXML

Permet de récupérer la valeur de tags dans un fichier XML, ou bien la valeur d'un attribut.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('PARSEXML')
CHGVAR          VAR(&PARM1) VALUE('
XMLFile="Fichier XML"
;Tags="Liste de tags ou 1 seul tag"
[;Attribut="Nom attribut"]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>XMLFile : Chemin complet et nom du fichier XML.</p> <p>Tags: liste des tags dont on veut récupérer les valeurs, séparés par des ';'.</p> <p>Attribut : nom de l'attribut du tag dont on veut récupérer la valeur. Si on cherche la valeur d'un attribut, il ne faut spécifier qu'un seul tag.</p> |
| RESULT | <p><u>En retour</u>:</p> <p>Liste des valeurs des tags passés en paramètres, séparées par des ';'.</p> <p>Si le paramètre Attribut est spécifié, on aura la valeur de l'attribut en retour.</p> |

Exemple 1

Supposons qu'on ait le XML suivant et qu'on veuille récupérer les valeurs suivantes (en rouge) :

```
<?xml version="1.0" encoding="utf-8"?>
<tnsb:Notification>
  <tnsb:NotificationType>GENERAL</tnsb:NotificationType>
  <tnsb:Request>
```

```

<tnsb:Type>PAPER</tnsb:Type>
<tnsb:TrackId>test_C33012646115</tnsb:TrackId>
<tnsb:DepositId>25527400032</tnsb:DepositId>
<tnsb:ReceptionDate>01/10/2015 1:17:34</tnsb:ReceptionDate>
<tnsb:FoldsCount>1</tnsb:FoldsCount>
<tnsb>Status>ACCEPT</tnsb>Status>
<tnsb:PaperOptions>
  <tnsb:EnvelopeType>C6</tnsb:EnvelopeType>
  <tnsb:PrintDuplex>>false</tnsb:PrintDuplex>
  <tnsb:DocumentCount>1</tnsb:DocumentCount>
  <tnsb:BilledPageCount>2</tnsb:BilledPageCount>
  <tnsb:PageCount>2</tnsb:PageCount>
</tnsb:PaperOptions>
</tnsb:Request>
</tnsb:Notification>
    
```

Si on exécute la commande suivante :

```

LNCCMDR    CMD(PARSEXML) +
           PARM1('XMLFILE="C:\A\notification.xml"; +
Tags="tnsb:Notification.tnsb:Request.tnsb:TrackId;tnsb:N+
           otification.tnsb:Request.tnsb:ReceptionDate+
           ;tnsb:Notification.tnsb:Request.tnsb>Status')
RESULT(&RESULT)
    
```

Alors en retour on aura :

```
*MSGtest_12;01/10/2015 1:17:34;ACCEPT
```

Exemple 2

Supposons qu'on ait le XML suivant et qu'on veuille récupérer les valeurs suivantes (en rouge) :

```

<?xml version="1.0" encoding="UTF-8"?>
<records>
  <record name="export" type="extra" date="20200402">
    <produits quantity="8" name="CatA">
      <produit name="A1" value="AS1254"/>
      <produit name="A2" value="UIPLH"/>
      <produit name="A3" value="125698YT"/>
      <produit name="A4" value="KJZ745"/>
      <produit name="A5" value="POJ523"/>
      <produit name="A6" value="127YUI"/>
      <produit name="A7" value="RTFG58"/>
      <produit name="A8" value="GHUP263"/>
    
```

Il faudra exécuter les commandes suivantes :

```

LNCCMDR    CMD(PARSEXML) +
           PARM1('XMLFILE="C:\test\produits.xml"+
           ;Tags="records.record";Attribut="type") +
RESULT(&RESULT)
    
```

```
LNCCMDR  CMD(PARSEXML) +  
          PARM1('XMLFILE="C:\test\produits.xml"+  
          ;Tags="records.record.produits.produit[6]"+  
          ;Attribut="value") RESULT(&RESULT)
```

Veillez noter que l'indice des tags commence à 0. Pour avoir le septième produit, on met donc l'indice = 6.

Commande PDFALLXL

Permet de créer un PDF pour chaque feuille d'un classeur Excel passé en paramètre. Chaque PDF aura le nom de la feuille Excel suivi de l'extension ".PDF".

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('PDFALLXL')
CHGVAR          VAR(&PARM1) VALUE('XL="Chemin classeur Excel"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | XL : chemin complet du classeur Excel. |

Exemple

```
LNCCMD          CMD(PDFALLXL) PARM1('XL="c:\A\test.xlsx"')
```

Commande PDFCONCAT

Permet de concaténer plusieurs fichiers PDF pour créer un nouveau fichier PDF.

Le nombre maximum de PDF à concaténer avec la commande PDFCONCAT est de 104.

Vous serez sans doute obligé de changer la taille de Parm1 (voir Exemple 2).

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('PDFCONCAT')
CHGVAR          VAR(&PARM1) VALUE('<path_file1>.pdf";"<path_file2>.pdf";
                "<path_file3>.pdf";"<path_file4>.pdf"')
CHGVAR          VAR(&PARM2) VALUE('Dest="Chemin PDF final"')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Chemin complet (entre guillemets) des différents fichiers PDF à concaténer, séparés par des ';'. |
| Parm2 | Dest : chemin complet du PDF final, correspondant au résultat de la concaténation des différents fichiers PDF contenus dans Parm1. |

Exemple 1

```
LNCCMD          CMD(PDFCONCAT) +
                PARM1('C:\A\a.pdf";"C:\A\b.pdf";"C:\A\R_27+
                WORD.pdf";"C:\A\c.pdf"') +
                PARM2('Dest="C:\A\res2.pdf"')
```

Exemple 2

Vous pouvez utiliser la commande DIRLIST pour lister les PDF d'un répertoire, et les envoyer en paramètre de PDFCONCAT.

Dans cet exemple, la commande PROPERTY a été utilisée pour changer la taille des paramètres Parm1, Parm2 et Result, seulement pour la connexion en cours. A la fin de la connexion, les paramètres par défaut sont appliqués.

PROPERTY n'a pas d'effet quand on utilise les commandes LNCCMD. Il faut utiliser l'appel de programme CALL PGM(LNCCMD).

Il faudra changer bien entendu la taille des variables.

La taille cumulée pour Parm1 et Parm2 ne doit pas dépasser 32700.
La taille pour Result ne doit pas dépasser 32700.

```

PGM
DCL          VAR(&RES)  TYPE(*CHAR)  LEN(512)
DCL          VAR(&OPT)  TYPE(*CHAR)  LEN(1)
DCL          VAR(&HDL)  TYPE(*CHAR)  LEN(100) VALUE('*ONLY ')
DCL          VAR(&PARM1) TYPE(*CHAR)  LEN(31000)
DCL          VAR(&PARM2) TYPE(*CHAR)  LEN(1000)
DCL          VAR(&LIST) TYPE(*CHAR)  LEN(20000)
DCL          VAR(&CNT)  TYPE(*DEC)  LEN(3) VALUE(0)
DCL          VAR(&REP)  TYPE(*CHAR)  LEN(3000)

LNCOPEN

LNCCMD       CMD(PROPERTY) +
              PARM1('PARMSIZE(31000,1000,512)')

CHGVAR       VAR(&REP) VALUE('C:\A\concat')
CHGVAR       VAR(&LIST) VALUE(' ')
CHGVAR       VAR(&PARM1) VALUE('PATTERN="*.pdf";PATH="' +
              *TCAT &REP *TCAT ' "')
CHGVAR       VAR(&PARM2) VALUE('FIRST=TRUE')

CALL         PGM(LNCCMD) PARM(&HDL 'DIRLIST' &OPT &PARM1 +
              &PARM2 &RES)

IF           COND(&RES *NE ' ') THEN(DO)
  CHGVAR     VAR(&LIST) VALUE(&LIST *TCAT ' "' *TCAT &REP +
              *TCAT '\ ' *TCAT &RES *TCAT ' "')
  CHGVAR     VAR(&CNT) VALUE(&CNT + 1)
ENDDO

CHGVAR       VAR(&PARM2) VALUE('FIRST=FALSE')

DOWHILE     COND(&RES *NE ' ')

  IF         COND(&CNT *GT 1) THEN(DO)
    CHGVAR   VAR(&LIST) VALUE(&LIST *TCAT ';" ' *TCAT &REP +
              *TCAT '\ ' *TCAT &RES *TCAT ' "')
  ENDDO

  CHGVAR     VAR(&CNT) VALUE(&CNT + 1)

  CALL       PGM(LNCCMD) PARM(&HDL 'DIRLIST' &OPT &PARM1 +
              &PARM2 &RES)
ENDDO

CHGVAR       VAR(&PARM1) VALUE(&LIST)
CHGVAR       VAR(&PARM2) +
              VALUE('DEST="C:\TEMP\FINAL_CONCAT.PDF"')
CALL         PGM(LNCCMD) PARM(&HDL 'PDFCONCAT' &OPT +
              &PARM1 &PARM2 &RES)
LNCCLOSE
    
```

ENDPGM



Commande PDFCOUNT

Permet de connaître le nombre de pages d'un PDF existant.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('PDFCOUNT')
CHGVAR          VAR(&PARM1) VALUE('File="Chemin PDF existant"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | File : chemin complet du PDF dont on veut connaître le nombre de pages. |
| RESULT | En retour de la commande, le nombre de pages est disponible sous la forme : PageCount=0000n. <u>Exemple :</u> PageCount=00002 |

Exemple

```
LNCCMDR        CMD(PDFCOUNT) PARM1('File="c:\A\b.pdf"') +
                RESULT(&RESULT)
```

La variable &RESULT contiendra la valeur suivante : PageCount=00002.

Commande PDFENCRYPT

Protège un document PDF existant.

L'encodage AES 256 est utilisé.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFENCRYPT')
CHGVAR          VAR(&PARM1) VALUE('
File=Fichier PDF à protéger"
;OwnerPWD=Mot de passe propriétaire"
[;UserPWD="Mot de passe utilisateur"]
[;EnablePrint= True / False ]
[;EnableCopy= True / False]
[;EnableEdit= True / False]
[;EnableNotes= True / False]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

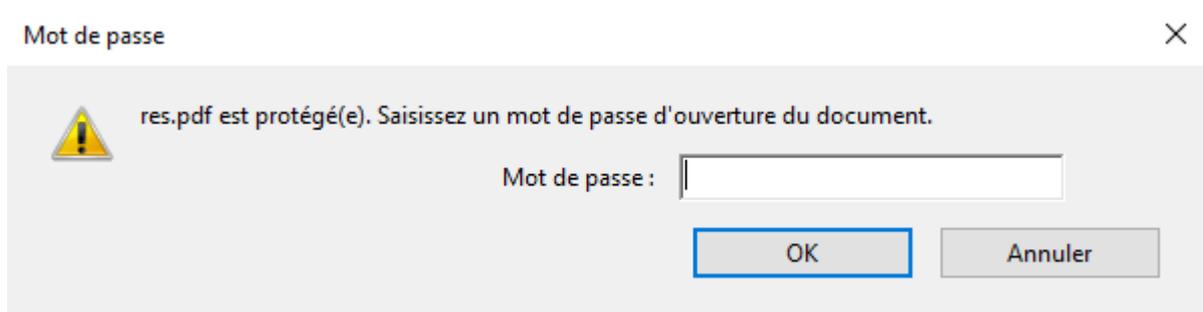
| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>File : Chemin complet et nom du fichier PDF à protéger.</p> <p>OwnerPWD: Mot de passe du propriétaire (Obligatoire). Avec ce mot de passe, le propriétaire pourra effectuer toute les opérations sur le document.</p> <p>UserPWD: Mot de passe utilisateur (Facultatif). Ce mot de passe sera demandé lors de toute ouverture du document.</p> <p>EnablePrint: Autoriser l'impression du document par les utilisateurs. Facultatif.</p> <p>EnableCopy: Autoriser la copie de texte ou de graphiques par les utilisateurs. Facultatif.</p> <p>EnableEdit: Autoriser la modification du document par les utilisateurs. Facultatif.</p> <p>EnableNotes: Autoriser les utilisateurs à ajouter des notes. Facultatif.</p> |

Exemple

Pour protéger un document en ajoutant un mot de passe propriétaire et utilisateur, on peut utiliser la commande suivante. veuillez noter qu'on autorise l'impression.

```
LNCCMD      CMD (PDFENCRYPT) +  
            PARM1 ('FILE="C:\temp\  
            res.pdf";OwnerPWD="ced";UserPWD="aura";E+  
            nablePrint=true')
```

L'ouverture du PDF nécessitera un mot de passe (UserPWD) :



Le PDF aura de plus les propriétés suivantes :

Protection du document

Méthode de protection : Protection par mot de passe

Mot de passe d'ouverture : Oui

Mot de passe d'accès aux droits : Oui

Impression : Haute résolution

Modification du document : Non autorisé(e)

Commentaires : Non autorisé(e)

Remplissage ou signature de formulaires : Non autorisé(e)

Assemblage du document : Non autorisé(e)

Copie du contenu : Non autorisé(e)

Accessibilité au contenu activée : Non autorisé(e)

Extraction de pages : Non autorisé(e)

Niveau de chiffrement : AES à 256 bits

OK

Commande PDFEXTRACT

Permet d'extraire certaines pages d'un PDF existant pour créer un nouveau fichier PDF.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFEXTRACT')
CHGVAR          VAR(&PARM1) VALUE('
                Dest="Chemin nouveau PDF";
                Origin="Chemin PDF original";
                StartPage=numéro de la 1ere page à extraire;
                EndPage=numéro de la dernière page à extraire
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>Dest : chemin complet du nouveau PDF, contenant certaines pages du PDF original.</p> <p>Origin : chemin complet du PDF original duquel on veut extraire certaines pages.</p> <p>StartPage : numéro de la 1ere page du PDF original à extraire.</p> <p>EndPage : numéro de la dernière page du PDF original à extraire.</p> |

Exemple

```

LNCCMD          CMD(PDFEXTRACT) +
                PARM1('Dest="c:\A\res_5.pdf";Origin="C:\A\r+
                es.pdf";StartPage=1;EndPage=2')
    
```

Le PDF « res_5.pdf » contiendra les pages 1 et 2 du PDF « res.pdf ».

Commande PDFFACTX

Permet de créer une facture électronique Factur-X.

Factur-X est un standard franco-allemand de facture électronique mixte, basé sur un fichier PDF (norme PDF/A3) représentant la facture originale et embarquant un fichier de données structuré (XML).

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFFACTX')
CHGVAR          VAR(&PARM1) VALUE('
                PDF="Chemin complet du PDF/A3"
                ;Xml="Chemin complet du fichier XML"
                ;XMP="Chemin complet du fichier XMP"
                ;Relationship="constante";
                ;Description="description Factur-X";
                ;FacturX="Chemin complet du fichier Factur-X à sauvegarder"
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>PDF: Chemin complet du PDF/A3.</p> <p>Xml: Chemin complet du fichier XML à insérer en pièce jointe.</p> <p>XMP : Chemin du fichier XML contenant les Meta data XMP du PDF/A.</p> <p>Relationship : Relation des données entre le fichier XML en pièce jointe et le PDF/A. Peut prendre une des valeurs suivantes (attention à respecter la casse) : Data, Source, Alternative, Supplement ou Unspecified.</p> <p>Description : Description Factur-X.</p> <p>FacturX : Chemin complet du fichier Factur-X à sauvegarder.</p> |

Remarque

Si on veut créer le PDF/A3 à partir d'un document Word, on peut utiliser la commande [WSAVEAS](#) avec Format=PDF et PdfA=true.

Voir l'exemple qui suit.

Exemple

PGM

```
DCL          VAR(&FILE) TYPE(*CHAR) LEN(2000)
DCL          VAR(&PATH) TYPE(*CHAR) LEN(2000)

CHGVAR      VAR(&PATH) VALUE('C:\temp\5')
CHGVAR      VAR(&FILE) VALUE('Doc2.docx')
LNCOPIEN    SVRADDR('*DEV')
LNCCMD      CMD(WORDOPEN)

LNCCMD      CMD(WOPENFILE) PARM1('File="" *TCAT &PATH +
          *TCAT '\'' *TCAT &FILE *TCAT ''')
LNCCMD      CMD(WSAVEAS) +
          PARM1('File="C:\temp\5\Doc3.pdf";Forma+
          t=PDF;PdfA=true')
LNCCMD      CMD(WORDCLOSE) PARM1('SAVE=FALSE')
LNCCMD      CMD(PDFFACTX) +
          PARM1('PDF="C:\temp\5\Doc+
          3.pdf";Xml="C:\temp\5\fa+
          ctur-x.xml";FacturX="C:\temp+
          \5\res29.PDF";DESCRIPTION="AURA +
          FACTUR-X";RELATIONSHIP="Data";XMP="C:\temp+
          \5\metadata_2023-22027.xml+
          ''')

LNCCLOSE
ENDPGM
```

Commande PDFFRIM

Permet de créer un PDF à partir d'une image.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('PDFFRIM')
CHGVAR          VAR(&PARM1) VALUE('
Image="Chemin complet du fichier image"
;Dest="Chemin complet du fichier PDF"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | Image: Chemin complet du fichier image dont le format peut être l'un des suivants : <ul style="list-style-type: none">- jpeg- bmp- png- gif- tiff Dest: Chemin complet du fichier PDF à créer. |

Exemple

```
LNCCMD          CMD(PDFFRIM) +
                 PARM1('Image="C:\temp\scanB.gif";Dest="C:\t+
emp\scanC.pdf"')
```

Commande PDFFRXL

Permet de créer un PDF à partir de la zone d'impression définie dans le classeur Excel courant.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('PDFFRXL')
CHGVAR          VAR(&PARM1) VALUE('
                File="Chemin complet du fichier PDF"
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | File: Chemin complet du fichier PDF à créer. Ce PDF sera basé sur la zone d'impression du classeur Excel courant. |

Exemple

Définition de la zone d'impression et création du PDF.

```
LNCCMD CMD(XLGOTOSH) PARM1('Sheet2')

LNCCMD CMD(XLCELLS)
PARM1('Ref="$A$1";RefTo="$J$27";Name="Print_Area"')

LNCCMD CMD(PDFFRXL) PARM1('File="C:\temp\resPDFFRXL.pdf"')
```

Commande PDFMERGE

Permet de superposer ou concaténer deux fichiers PDF.

Cette commande peut être utilisée pour ajouter un filigrane à un document PDF existant. Elle utilise LAUNCHER_PDF.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFMERGE')
CHGVAR          VAR(&PARM1) VALUE('
                File="Fichier de sortie"
                ;Source1="Document PDF No 1"
                ;Source2="Document PDF No 2"
                [;Directory="Répertoire de destination"]
                [;Repeat= True / False]
                [;Above= True / False]
                [;Append= True / False]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>File : Chemin complet et nom du fichier PDF résultat de la fusion ou concaténation, qui sera effectuée entre les deux fichiers désignés par "Source1" et "Source2".</p> <p>Source1 : Chemin complet et nom du premier fichier PDF original.</p> <p>Source2 : Chemin complet et nom du deuxième fichier PDF, qui servira de filigrane pour le document résultat, ou qui sera ajouté à la fin de Source1 si Append est vrai.</p> <p>Repeat : Si Repeat est vrai, et Above est vrai, le document Source2 sera répété sur toutes les pages de Source1. Sinon, le résultat n'aura un filigrane que sur la première page.</p> <p>Above : Vrai pour que le contenu de Source2 apparaisse en filigrane de Source1.</p> <p>Append : Vrai pour ajouter le contenu de Source2 à la suite de Source1. Dans ce cas, il n'y aura pas de filigrane ajouté au résultat.</p> |

Remarque

Si Above et Append ne sont pas spécifiés, c'est la valeur True par défaut de Above qui sera prise en compte. On procédera donc à la mise en place d'un filigrane.

Si Append=true alors on procédera à la concaténation, quelque soit la valeur de Above.

Exemple 1

Fabrique un PDF temporaire avec le mot "DUPLICATA" en filigrane.
Fusionne ce PDF avec un PDF original, avec répétition du filigrane sur toutes les pages.

```
/*ouvre Word avec une page blanche */
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('New')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/* Prépare le driver PDF avec un filigrane "DUPLICATA" */
CHGVAR VAR(&CMD) VALUE('PDFPRINTER')
CHGVAR VAR(&PARM1) VALUE('File="%TEMP%\filigrane.pdf" +
WMText="DUPLICATA"')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/* Imprime la page blanche vers un PDF temporaire */
CHGVAR VAR(&CMD) VALUE('WPRINT')
CHGVAR VAR(&PARM1) VALUE('Printer="LAUNCHER_PDF"')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/* fusionne "Original" avec "filigrane" */
CHGVAR VAR(&CMD) VALUE('PDFMERGE')
CHGVAR VAR(&PARM1) VALUE('File="C:\documents\resultat.pdf"; +
Source1="C:\Documents\original.pdf"; +
Source2="%TEMP%\filigrane.pdf"; +
Repeat=True')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
/* "C:\documents\resultat.pdf" contient "DUPLICATA"
sur toutes les pages */
```

Exemple 2



Concaténation des PDF a.pdf et b.pdf pour créer le PDF res_con.pdf.

```
LNCCMD      CMD (PDFMERGE) +  
            PARM1 ('FILE="C:\A\res_con.pdf";SOURCE1="C:\+  
            A\a.pdf";SOURCE2="C:\A\b.pdf";APPEND=TRUE')
```

Commande PDFORDERX

Permet de créer un bon de commande électronique Order-X.

Basé sur Factur-X, Order-X est un standard franco-allemand de bon de commande électronique mixte, basé sur un fichier PDF (norme PDF/A3) représentant le bon de commande original et embarquant un fichier de données structuré (XML).

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFORDERX')
CHGVAR          VAR(&PARM1) VALUE('
PDF="Chemin complet du PDF/A3"
;Xml="Chemin complet du fichier XML"
;XMP="Chemin complet du fichier XMP"
;RelationShip="constante";
;Description="description Order-X";
;OrderX="Chemin complet du fichier Order-X à sauvegarder"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>PDF: Chemin complet du PDF/A3.</p> <p>Xml: Chemin complet du fichier XML à insérer en pièce jointe.</p> <p>XMP : Chemin du fichier XML contenant les Meta data XMP du PDF/A.</p> <p>RelationShip : Relation des données entre le fichier XML en pièce jointe et le PDF/A. Peut prendre une des valeurs suivantes (attention à respecter la casse) : Data, Source, Alternative.</p> <p>Description : Description Order-X.</p> <p>OrderX : Chemin complet du fichier Order-X à sauvegarder.</p> |

Remarque

Si on veut créer le PDF/A3 à partir d'un document Word, on peut utiliser la commande [WSAVEAS](#) avec Format=PDF et PdfA=true. Voir l'exemple qui suit.

Exemple

PGM

```
DCL          VAR(&FILE) TYPE(*CHAR) LEN(2000)
DCL          VAR(&PATH) TYPE(*CHAR) LEN(2000)

CHGVAR      VAR(&PATH) VALUE('C:\temp\5')
CHGVAR      VAR(&FILE) VALUE('order.docx')
LNCOPIEN    SVRADDR('*DEV')
LNCCMD      CMD(WORDOPEN)

LNCCMD      CMD(WOPENFILE) PARM1('File="" *TCAT &PATH +
*TCAT '\'' *TCAT &FILE *TCAT '\''')
LNCCMD      CMD(WSAVEAS) +
          PARM1('File="C:\temp\5\Doc.pdf";Forma+
          t=PDF;PdfA=true')
LNCCMD      CMD(WORDCLOSE) PARM1('SAVE=FALSE')
LNCCMD      CMD(PDFORDERX) +
          PARM1('PDF="C:\temp\5\Doc+
          .pdf";Xml="C:\temp\5\or+
          der-x.xml";OrderX="C:\temp+
          \5\res.PDF";DESCRIPTION="AURA +
          ORDER-X";RELATIONSHIP="Data";XMP="C:\temp+
          \5\metadata_2024.xml+
          ''')

LNCCLOSE
ENDPGM
```

Commande PDFOVER

Permet d'ajouter un filigrane ou un fonds de page à un PDF existant pour créer un nouveau fichier PDF.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFOVER')
CHGVAR          VAR(&PARM1) VALUE('
                Dest="Chemin PDF final";
                Origin="Chemin PDF original";
                Overlay="Chemin PDF fonds de page"
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>Dest : chemin complet du PDF final, correspondant au PDF original auquel on a ajouté le fond de page ou filigrane.</p> <p>Origin : chemin complet du PDF original auquel on veut ajouter le fond de page ou filigrane.</p> <p>Overlay : chemin complet du PDF contenant le fond de page ou filigrane.</p> |

Exemple

```

LNCCMD          CMD(PDFOVER) +
                PARM1('Dest="c:\A\res_2.pdf";Origin="C:\A\at+
                .pdf";Overlay="C:\A\over.pdf"')
    
```

Commande PDFPRINT

Imprime un document PDF existant, vers une imprimante.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFPRINT')
CHGVAR          VAR(&PARM1) VALUE('
File="Fichier à imprimer"
;Printer="Nom de l'imprimante vers laquelle imprimer"
[PWD="Mot de passe éventuel"]
[;StartPage= Nombre ]
[;EndPages= Nombre]
[;Copies= Nombre ]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>File : Chemin complet et nom du fichier PDF à imprimer.</p> <p>Printer: Nom de l'imprimante vers laquelle imprimer le document.</p> <p>PWD: Si le document est protégé, le mot de passe doit être spécifié ici.</p> <p>StartPage: Numéro de la première page à imprimer. Par défaut, le document est imprimé à partir de la page 1</p> <p>EndPage: Numéro de la dernière page à imprimer. Par défaut le document est imprimé jusqu'à la fin.</p> <p>Copies: Nombre d'exemplaires à imprimer. Par défaut le nombre d'exemplaire est égal à 1.</p> |

Exemple

```

CHGVAR          VAR(&CMD) VALUE('PDFPRINT')
CHGVAR          VAR(&PARM1) VALUE('File="C:\Documents\Resultat.pdf"')
CHGVAR          VAR(&PARM2) VALUE('Printer="\\Server\HP Laser 2200"')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Voir aussi

- [Commande PDFPRINTER](#)

Commande PDFPRINTER

Permet de définir les options de génération du fichier PDF par le driver d'impression LAUNCHER_PDF.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFPRINTER')
CHGVAR          VAR(&PARM1) VALUE('
File="Fichier de sortie"
[;Directory=Répertoire de destination]
[;Resolution=Nombre]
[;PaperSize= Nombre]
[;PaperWidth= Nombre]
[;PaperLength= Nombre]
[;Orientation= Nombre]
[;Append= True / False]
[;EmbedFonts= True / False]
[;WMText= "Texte en filigrane"]
[;WMFont= "Police pour le filigrane"]
[;WMSize= Nombre]
[;WMOrientation= Nombre]
[;WMHPos= Nombre]
[;WMVPos= Nombre]
[;WMColor= Nombre]
[;WMForeground= True / False]
[;JPEGCompress= True / False]
[;JPEGLeval= Nombre]
[;Default= True / False]
[;Encrypt= True / False]
[;HyperLinks= True / False]
[;ScalingOption= 1 / 0]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>File : Nom du fichier PDF à générer. Le chemin des répertoires peut être inclus dans cette valeur.</p> <p>Directory: Chemin des répertoire vers le fichier PDF à générer. Ce chemin peut être absent s'il est spécifié avec la propriété File.</p> <p>Resolution : Résolution de sortie en DPI. Les valeurs possibles sont : 75, 150, 300, 600, 1200.</p> <p>PaperSize : Format de la page.</p> |

Les valeurs possibles sont : 1=Letter ; 5=legal ; 9=A4 ; 8=A3 ; 256=personnalisé.

PaperWidth : Largeur de page en dixième de millimètres.

PaperLength : Hauteur de page en dixième de millimètres.

Orientation: Orientation. 1=Portrait; 2=Paysage.

Append: Vrai pour ajouter le nouveau document à la fin du fichier PDF si celui ci existe déjà.
Faux par défaut.

EmbedFonts: Vrai pour embarquer les polices dans le document PDF. Faux par défaut.

WMText : Texte à imprimer en filigrane.

WMFont : Nom de la police pour le filigrane (Défaut=Arial).

WMSize : Taille en 1/100 pouces (Défaut=70).

WMOrientation : Orientation en 1/10 degrés (Défaut=3150).

WMHPos, WMVPos : Position Horizontale et Vertical en 1/100 pouces (Défaut= 100, 500).

WMColor : Couleur en RGB (Défaut=12632256).

WMForeground : Vrai pour que le filigrane soit en premier plan.

JPEGCompress : Vrai pour activer la compression des images JPEG.

JPEGLLevel peut prendre les valeurs:
3 pour une compression forte,
7 pour une compression moyenne,
9 pour une compression faible.

Default : Vrai pour faire de l'imprimante PDF, l'imprimante par défaut.

Encrypt : Vrai pour protéger le document. Un document protégé ne peut pas être imprimé ni modifié.

HyperLinks : Vrai pour transformer en lien hyper texte, les textes commençant par "**http:**" ou "**www.**".

ScalingOption : Si ScalingOption = 1, le contenu est ajusté à la taille du PDF (étendu ou réduit). Valeur par défaut =0 : pas d'ajustement.

Remarque

La commande PDFPRINTER initialise le driver d'impression LAUNCHER_PDF.
La commande d'impression qui suit l'appel de la commande PDFPRINTER doit indiquer que l'imprimante de destination est « LAUNCHER_PDF ».

Un filigrane est ajouté sur chaque page si **WMText** est fixé.
La couleur **WMCOLOR** est exprimée en niveaux de Rouge, Vert, Bleu, selon la formule :

$((\text{Bleu} * 256) + \text{Vert}) * 256 + \text{Rouge}$.

Exemple 1

Prépare la génération d'un fichier PDF vers le fichier "temp.pdf" du répertoire temporaire.
Le mot DUPLICATA sera imprimé en filigrane sur chaque page.
Imprime le document Word avec l'imprimante LAUNCHER_PDF pour le convertir en PDF.

```
PGM
```

```
LNCOPEN
```

```
LNCCMD CMD (WORDOPEN)
```

```
LNCCMD CMD (WOPENFILE) PARM1 ('C:\PROGRAM FILES\LAUNCHER400\+  
SAMPLES\LNCHELL.DOC') PARM2 (VISIBLE)
```

```
LNCCMD CMD (PDFPRINTER) +  
PARM1 ('FILE="C:\TEMP\LNCHELL.PDF";wmtxt+  
="DUPLICATA"')
```

```
LNCCMD CMD (WPRINT) PARM1 ('PRINTER="LAUNCHER_PDF"')
```

```
LNCCMD CMD (WORDCLOSE)
```

```
LNCSHELL CMD ('C:\TEMP\LNCHELL.PDF') WAITCMD (*NO) +  
VISIBLE (*YES) MINCURWIN (*YES) +  
ACTION (OPEN) EXESRV (*CURRENT)
```

```
ENDPGM
```

Exemple 2

Ouvrir un fichier Excel et le convertir en PDF :

```
PGM
```

```
LNCOOPEN
```

```
LNCCMD CMD (EXCELOPEN) PARM1 ('')
```

```
LNCCMD CMD (XLOPENFILE) +  
PARM1 ('C:\TEMP\MIGRATIONAVEC.XLS')
```

```
LNCCMD CMD (PDFPRINTER) +  
PARM1 ('FILE="C:\TEMP\MIGRATIONAVEC.PDF"')
```

```
LNCCMD CMD (XLPRINT) PARM1 ('PRINTER="LAUNCHER_PDF"')
```

```
LNCCMD CMD (EXCELCLOSE)
```

```
LNCSHELL CMD ('C:\TEMP\MIGRATIONAVEC.PDF') +  
WAITCMD (*NO) VISIBLE (*YES) +  
MINCURWIN (*YES) ACTION (OPEN) EXESRV (*CURRENT)
```

```
ENDPGM
```

Voir aussi

- [Commande PDFPRINT](#)

Commande PDFSEARCH

Permet de faire une recherche (sensible à la casse) de texte dans le contenu du PDF.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('PDFSEARCH')
CHGVAR          VAR(&PARM1) VALUE('
File="Chemin fichier PDF";
Text=texte à rechercher
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | File : chemin complet du fichier PDF (entre guillemets) dans lequel on veut rechercher du texte. Text : texte à chercher, sans guillemets. |
| RESULT | En retour : <ul style="list-style-type: none"> - true si le texte est trouvé dans le contenu du PDF - false : si le texte n'est pas trouvé dans le contenu du PDF |

Exemple

```
LNCCMDR        CMD(PDFSEARCH) +
                PARM1('File="c:\A\res.pdf";Text=Erica') +
                RESULT(&RESULT)
```

Commande PDFSIGN

Permet de signer un PDF existant avec un certificat au format PFX/P12/PKCS#12.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFSIGN')
CHGVAR          VAR(&PARM1) VALUE('
File="Fichier PDF à signer"
;Certificate="Chemin complet du fichier contenant le
certificat"
;SaveAs="Chemin complet du fichier PDF signé"
;PWD="Mot de passe du certificat"
;SignedBy="Nom qui apparaîtra sur la signature"
;Reason="Motif qui apparaîtra sur la signature"
;Location="Lieu qui apparaîtra sur la signature"
[;ImageFile= "Chemin complet de l'image JPG qui apparaîtra
sur le PDF"]
[;PageNumber= Numéro de la page où sera apposée la signature
]
[;HorzPos= Position horizontale de la signature sur la
feuille]
[;VertPos= Position verticale de la signature sur la feuille]
[;Zoom= Exprimé en % par rapport à la taille de l'image
spécifiée ]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>File : Fichier PDF à signer</p> <p>Certificate: Chemin complet du fichier contenant le certificat au format PFX/P12/PKCS#12.</p> <p>PWD: Mot de passe du certificat.</p> <p>SaveAs: Chemin complet du fichier PDF signé.</p> <p>SignedBy: Nom qui apparaîtra sur la signature.</p> <p>Reason: Motif qui apparaîtra sur la signature.</p> <p>Location : Lieu qui apparaîtra sur la signature.</p> |

ImageFile : Chemin complet de l'image qui apparaîtra sur le PDF (facultatif). Format JPG obligatoire.

PageNumber : Numéro de la page où sera apposée la signature (facultatif). Par défaut la signature apparaîtra sur la page 1.

HorzPos : Position horizontale de la signature sur la feuille (facultatif). Par défaut HorzPos = 0.

VertPos : Position verticale de la signature sur la feuille (facultatif). Par défaut VertPos = 0.

Zoom : Exprimé en % par rapport à la taille de l'image spécifiée (facultatif). Une valeur positive augmentera la taille de l'image, alors qu'une valeur négative la diminuera. Par défaut Zoom = -50.

Exemple

```
LNCCMD      CMD (PDFSIGN) +  
             PARM1 ('FILE="C:\A\PDF_test\resultat.pdf";Sa+  
             veAs="C:\A\PDF_test\resultat_IM2.pdf";Signe+  
             dBy="AURA";Reason="Approve";Location="NICE"+  
             ;certificate="C:\temp\aura.p12";PWD="aura";+  
             ImageFile="C:\temp\logo.jpg"')
```

Commande PDFSPLIT

Permet de splitter un PDF existant en plusieurs fichiers PDF.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFSPLIT')
CHGVAR          VAR(&PARM1) VALUE('<path_file1>.pdf";"<path_file2>.pdf";
                "<path_file3>.pdf";"<path_file4>.pdf"')
CHGVAR          VAR(&PARM2) VALUE('
                Origin="Chemin PDF original";
                SplitPagesQty=nombre de pages par PDF splitté
                ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Chemin complet des différents nouveaux fichiers PDF issus du split du PDF original, séparés par des ` `; |
| Parm2 | <p>Origin : chemin complet du PDF Original à splitter.</p> <p>SplitPagesQty : nombres de pages de chaque PDF issu du split du PDF original.</p> <p><u>Remarque</u> :</p> <p>Si le PDF original a 5 pages, et que SplitPagesQty =2 (on veut 2 pages par PDF splitté), on aura :</p> <ul style="list-style-type: none"> - 2 pages pour le 1^{er} PDF spécifié dans Parm1 - 2 pages pour le 2eme PDF spécifié dans Parm1 - 1 page pour le 3eme PDF spécifié dans Parm1 |

Exemple

```

LNCCMD          CMD(PDFSPLIT) +
                PARM1('C:\A\b_1.pdf";"C:\A\b_2.pdf"') +
                PARM2('Origin="C:\A\b.pdf";SplitPagesQty=1')
    
```

Commande PDFTOIM

Permet de créer une image au format JPG, pour chaque page d'un fichier PDF.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('PDFTOIM')
CHGVAR          VAR(&PARM1) VALUE('
                File="Chemin complet du fichier PDF"
                [;StartPage=numéro de la 1ere page à convertir]
                [;EndPage=numéro de la dernière page à convertir]
                [ ;PWD="Mot de passe éventuel"]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | File: Chemin complet du fichier PDF. |
| ou | StartPage : Numéro de la 1ere page du PDF à convertir. |
| Parm2 | EndPage : Numéro de la dernière page du PDF à convertir. |
| | PWD: Si le document est protégé, le mot de passe doit être spécifié ici. |

Remarque

Les fichiers JPG seront créés au même endroit que le fichier PDF.

Ils auront le même nom que le fichier PDF avec en suffixe, le numéro de la page.

Exemple : si le PDF se nomme "R_01WORD.PDF", alors les fichiers JPEG créés à côté du PDF, auront les noms suivants :

R_01WORD2.jpg, R_01WORD3.jpg, R_01WORD4.jpg...

Exemple

```

LNCCMD CMD(PDFTOIM) +
PARM1('File="C:\A\PDFTOIMG\R_01WORD.PDF";StartPage=2;EndPage=4')
    
```

Commande *PRINTPRNF*

Permet d'envoyer un fichier PRN vers une imprimante Windows.

Un fichier PRN contient un flux de données destiné à une imprimante (Ex: PCL). Ce flux est généré par une commande d'impression vers un fichier.

Syntaxe

```
CHGVAR VAR(&CMD) VALUE('PRINTPRNF')
CHGVAR VAR(&PARM1) VALUE('
    File="Chemin et nom du fichier PRN à imprimer";
    Printer=Nom de l'imprimante à utiliser";
    [ Document="Nom de document pour le pilote d'imprimante"; ]
    ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LAUNCHER/LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
    &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>File désigne le chemin et le nom du fichier d'impression à envoyer vers l'imprimante. Généralement ce fichier à un suffixe ".PRN". Il contient un flux d'impression. Ce fichier est généré par une commande d'impression vers un fichier.</p> <p>Printer désigne l'imprimante vers laquelle on veut envoyer le flux d'impression. L'imprimante doit être compatible avec le flux contenu dans le fichier d'impression. Par exemple : Le fichier d'impression a été généré avec un pilote d'impression PCL 5, alors, l'imprimante désignée ici doit être compatible PCL 5.</p> <p>Document est facultatif. Il permet d'indiquer un nom de document au pilote d'imprimante. Ce nom apparaîtra par exemple dans les files d'attente du spool Windows.</p> |

Exemple

```
CHGVAR VAR(&CMD) VALUE('WPRINT')
CHGVAR VAR(&PARM1) VALUE('Printer="HP LJ 1200";OutFile=
    "%TEMP%\Out.prn"')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 + &PARM2
```

```
&RESULT)

CHGVAR          VAR(&CMD) VALUE('PRINTPRNF')
CHGVAR          VAR(&PARM1)
                VALUE('File="%TEMP%\Out.prn";Printer="\\server\HP
                LJ 2200"')

CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 + &PARM2
                &RESULT)
```

Voir aussi

[Commande SETPRINTER](#)

Commande *PROPERTY*

Permet de changer des propriétés de LAUNCHER Office pour la connexion en cours.

On peut ainsi changer la taille des paramètres de la commande [LNCCMD - Programme API](#).

Syntaxe

```
CHGVAR VAR(&CMD) VALUE('PROPERTY')
CHGVAR VAR(&PARM1) VALUE('
    PARMsize(Taille_Parm1,Taille_Parm2,Taille_Result)
')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LAUNCHER/LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
    &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Propriété a changer pour la connexion LAUNCHER Office en cours.</p> <p>PARMSIZE(<i>Taille_Parm1,Taille_Parm2,Taille_Result</i>) permet de changer la taille par défaut des paramètres &Parm1, &Parm2 et &Result lors de l'appel du programme LNCCMD.</p> <p>Par défaut, les paramètres ont les tailles suivantes :</p> <ul style="list-style-type: none">• 512 pour Parm1,• 1024 pour Parm2,• 512 pour Result. <p>Si une taille donnée pour un paramètre est nulle, la propriété n'est pas modifiées pour ce paramètre.</p> <p>La taille cumulée pour <i>Parm1</i> et <i>Parm2</i> ne doit pas dépasser 32700. La taille pour <i>Result</i> ne doit pas dépasser 32700.</p> |

Remarque

PROPERTY n'a pas d'effet quand on utilise les commandes LNCCMD. Il faut utiliser l'appel de programme CALL PGM(LNCCMD).

La commande LNCCMD est défini avec 512 caractères pour PARM1. L'interpréteur de l'OS empêche donc l'exécution si supérieur.

En bref :

Ajouter les définitions suivantes en début de votre CL.

```
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50) VALUE('*ONLY')
DCL VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL VAR(&PARM1) TYPE(*CHAR) LEN(2000)
DCL VAR(&PARM2) TYPE(*CHAR) LEN(2000)
DCL VAR(&RESULT) TYPE(*CHAR) LEN(512)
```

Quand vous avez besoin d'envoyer plus de 512 caractères, utilisez :

CALL PGM(LNCCMD) et non la commande LNCCMD

Par exemple, dans votre code, remplacez :

```
LNCCMD CMD(FILEWRITE) PARM1(&FILEWRITE)
```

Par :

```
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &FILEWRITE &PARM2 &RESULT)
```

Exemple

```
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(256)
DCL          VAR(&PARM2) TYPE(*CHAR) LEN(256)
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)

CHGVAR      VAR(&CMD) VALUE('PROPERTY')
CHGVAR      VAR(&PARM1) VALUE('PARMSIZE(256,256,512)')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Voir aussi

[Déclaration des variables](#)

Commande RCL

Permet de rappeler une fenêtre mémorisée.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('RCL')
CHGVAR      VAR(&PARM1) VALUE('[Numéro de fenêtre]')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | "Numéro de fenêtre" est facultatif. Ce numéro a précédemment été retourné par la commande STO ou MIN . S'il est spécifié, c'est la fenêtre dont le numéro est donné qui sera restaurée, et rendue visible. S'il est absent, la dernière fenêtre sauvegardée par une commande STO sera restaurée. |

Voir aussi

- [MIN](#)
- [STO](#)

Commande *REMOVEDIR*

Permet de supprimer un répertoire vide sur le PC/Serveur Windows.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('REMOVEDIR')
CHGVAR          VAR(&PARM1) VALUE('
Dir="Répertoire à supprimer"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2
&RESULT)
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | Dir : Chemin complet et nom du répertoire à supprimer. |
| RESULT | Contient éventuellement le message d'erreur. |

Remarque

Le répertoire doit être vide, et des droits suffisants sont nécessaires pour le profil utilisé.

Exemple

```
LNCCMD CMD(REMOVEDIR) PARM1('Dir="C:\temp\DEL"')
```

Commande *RSTPRINTER*

Restaure les paramètres d'origine de l'imprimante spécifiée.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('RSTPRINTER')
CHGVAR          VAR(&PARM1) VALUE('Printer="Nom de l'imprimante"')

CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---------------------------------------|
| Parm1 | Printer = Nom de l'imprimante. |

Voir aussi

- [SETPRINTER](#)

Commande SAISI

Permet de créer un formulaire personnalisé.

Il est possible de spécifier le titre du formulaire et 7 champs avec un message d'aide pour chacun.

La commande SAISI est obligatoirement suivie de la commande [SAISIWAIT](#).

Pour marcher correctement, il est recommandé que le pare-feu soit activé, et que le programme Launcher (Incsrv.exe et Incadm.exe) soit dans la liste des programmes autorisés.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('SAISI')
CHGVAR          VAR(&PARM1) VALUE('
CAPTIONS="Libellés des champs"
[;TIPS="Aide affichée pour remplir chaque champ"]
[;FIELDNAME="Titre du formulaire"]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>CAPTIONS : libellés de chaque champ, séparés par des points virgules. Jusqu'à 7 champs possibles.</p> <p>TIPS : aide affichée pour chaque champ lorsqu'on approche la souris du champ de saisi. Le nombre de « tips » doit correspondre au nombre de "captions". Ils doivent être séparés par des points virgules. Optionnel.</p> <p>FIELDNAME : titre du formulaire. Optionnel.</p> |

Exemple

```

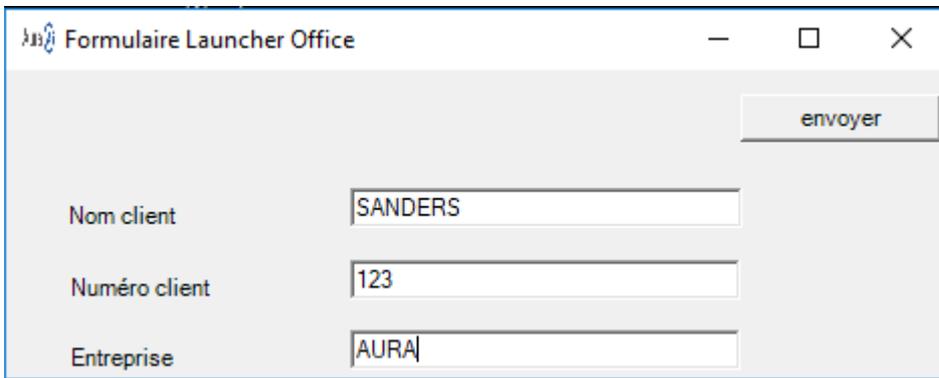
/* CREATION MENU */
CHGVAR          VAR(&CMD) VALUE('SAISI')
CHGVAR          VAR(&PARM1) VALUE('CAPTIONS="Nom +
client;Numéro +
client;Entreprise";TIPS="Ecrire le nom +
du client;Ecrire le numéro client;Ecrire +
le nom entreprise";FIELDNAME="Formulaire +
Launcher Office"')
    
```



```
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                           &PARM2 &RESULT)

/* ATTENTE CLICK SUR BOUTON */
CHGVAR      VAR(&CMD) VALUE('SAISIWAIT')
CHGVAR      VAR(&OPT) VALUE(' ')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                           &PARM2 &RESULT2)
LNCCMD      CMD(NOP) PARM1(&RESULT2)
```

Le formulaire créé est le suivant:



| | |
|---------------|--------------------------------------|
| Nom client | <input type="text" value="SANDERS"/> |
| Numéro client | <input type="text" value="123"/> |
| Entreprise | <input type="text" value="AURA"/> |

envoyer

Une fois les champs remplis, l'utilisateur clique sur "envoyer", et les données saisies sont récupérées par le programme, dans la variable &RESULT2 de la commande SAISIWAIT. Les données de chaque champ sont séparées par des points virgules.

Dans notre exemple, la variable &RESULT2 contiendra :

SANDERS;123;AURA

Commande SAISIWAIT

Permet de récupérer les données saisies dans le formulaire personnalisé créé par la commande SAISI.

La commande SAISIWAIT suit obligatoirement la commande [SAISI](#).

Syntaxe

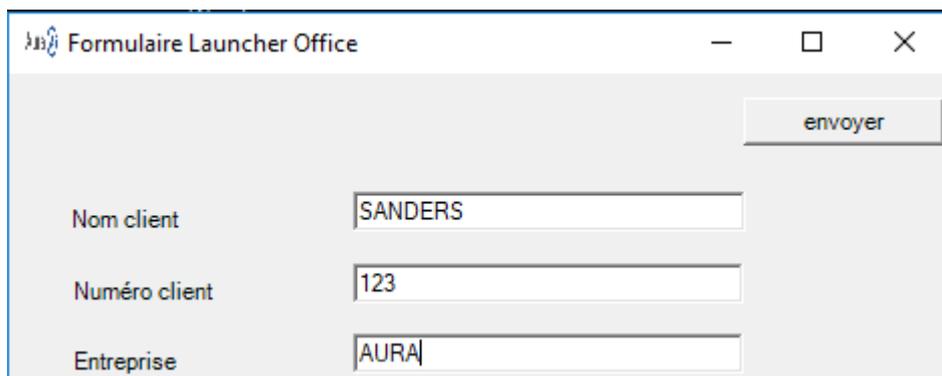
```
CHGVAR      VAR(&CMD) VALUE('SAISIWAIT')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Exemple

```
/* CREATION MENU */
CHGVAR      VAR(&CMD) VALUE('SAISI')
CHGVAR      VAR(&PARM1) VALUE('CAPTIONS="Nom +
client;Numéro +
client;Entreprise";TIPS="Ecrire le nom +
du client;Ecrire le numéro client;Ecrire +
le nom entreprise";FIELDNAME="Formulaire +
Launcher Office"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/* ATTENTE CLICK SUR BOUTON */
CHGVAR      VAR(&CMD) VALUE('SAISIWAIT')
CHGVAR      VAR(&OPT) VALUE(' ')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT2)
LNCCMD      CMD(NOP) PARM1(&RESULT2)
```

Le formulaire créé est le suivant:



Une fois les champs remplis, l'utilisateur clique sur "envoyer", et les données saisies sont récupérées par le programme, dans la variable &RESULT2 de la commande SAISIWAIT. Les données de chaque champ sont séparées par des points virgules.

Dans notre exemple, la variable &RESULT2 contiendra :

SANDERS;123;AURA

Commande SCAN

Permet de piloter un scanner et d'enregistrer le document scanné dans différents formats.

Le nom du scanner doit être celui retourné par la commande [LISTSCAN](#).

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('SCAN')
CHGVAR          VAR(&PARM1) VALUE('
Scanner="Nom du scanner"
;Type="Format dans lequel le document scanné sera enregistré"
;Dest="Chemin complet du fichier scanné"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>Scanner : Nom du scanner. Utilisez la commande LISTSCAN pour voir le nom du scanner à utiliser.</p> <p>Type: Format dans lequel le document scanné sera enregistré. Les différents formats possibles sont :</p> <ul style="list-style-type: none"> - jpeg - bmp - png - gif - tiff - pdf <p>Dest: Chemin complet du fichier scanné.</p> |

Exemple

```

LNCCMD          CMD(SCAN) PARM1('Scanner="HP LJ100 M175 +
Scan";Dest="C:\temp\scanB.gif";Type="gif"')

LNCCMD          CMD(SCAN) PARM1('Scanner="HP LJ100 M175 +
Scan";Dest="C:\temp\scanC.pdf";Type="pdf"')
    
```

Commande *SELECTFILE*

Ouvre la boîte de dialogue de sélection d'un fichier.

Le chemin complet du fichier sélectionné est retourné dans la variable &RESULT.

Si l'utilisateur annule la sélection on récupère la valeur *CANCEL.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE ('SELECTFILE')
CHGVAR          VAR(&PARM1) VALUE ('
                [Pattern="Fichiers recherchés"]
                [; File="Nom du Fichier recherché" ]
                [; Title="Titre de la fenêtre" ]
                [; Directory="Répertoire initial" ]
                [; PathMustExist ]
                [; FileMustExist ]
                [; CreatePrompt ]
                [; OverWritePrompt ]
                [; NameOnly ]
                [; DftExt="Extension par défaut" ]
                [; Save ]
                [; Folders ]
                [; Copy ]
                ')
CHGVAR          VAR(&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Pattern donne la liste des catégories de fichiers recherchés. Chaque catégorie est représentée par un texte de description, suivi du caractère "=", et suivi de la liste des types séparés par un point virgule. Les catégories sont séparées entre elles par un double point virgule.</p> <p>Exemple : Pattern="Fichiers textes=*.doc ;*.dot ;*.txt ;;Tous=*.*" </p> <p>Si Pattern n'est pas précisé, se sont les fichiers Word qui sont recherchés.</p> <p>File permet de donner un nom initial recherché.</p> <p>Title change le titre de la fenêtre de recherche.</p> |

Directory donne le répertoire initial de la recherche.

PathMustExist indique que le répertoire doit exister.

FileMustExist indique que le fichier doit exister.

CreatePrompt indique que l'on veut une confirmation pour la création du fichier si celui ci n'existe pas.

OverWritePrompt indique que l'on veut une confirmation du remplacement du fichier si celui existe déjà.

NameOnly indique que l'on ne veut que le nom du fichier en retour, sans le chemin complet.

DftExt permet de donner une extension par défaut, qui sera ajouté au nom choisi, si l'utilisateur n'a pas entré d'extension.

Save permet de choisir la boîte de dialogue de Windows dédiée à la recherche d'un fichier à sauver.

Folders indique que l'on veut parcourir les répertoires seulement.

Copy entraînera la copie du chemin et du nom du fichier sélectionné, vers le presse papier. L'utilisateur pourra les coller dans une autre application

Remarque

Lors de la saisie d'une commande CL de LAUNCHER Office tel que **LNCPRDOC**, en pressant **F4** sur les paramètres qui contiennent un "chemin vers un fichier", c'est cette commande **SELECTFILE** qui est appelée. L'option **Copy** est donnée, ce qui permet de coller le chemin depuis l'émulateur de terminal.

Exemple

```
CHGVAR          VAR(&CMD) VALUE ('SELECTFILE')
CHGVAR          VAR(&PARM1) VALUE (' +
Pattern="Text file=*.doc;*.dot;*.txt;;Excel
files=*.xls;;All files=*.*" ; +
Directory="C:\My Documents";FileMustExist')
```



```
CHGVAR          VAR (&PARM2) VALUE ( ' ' )  
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +  
                &PARM2 &RESULT)
```

Voir aussi

- [CHKFILE](#)
- [LNCPRDOC - Commande CL](#)

Commande SETPRINTER

Cette commande permet au programme AS/400 de:

- Modifier certaines propriétés de l'imprimante désignée.
- Configurer une imprimante Windows avec une configuration précédemment enregistrée.
- Rediriger la sortie de l'imprimante vers un fichier.
- Changer l'imprimante par défaut.

L'appel à la commande RSTPRINTER annule les changements effectués par SETPRINTER

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('SETPRINTER')
CHGVAR          VAR(&PARM1) VALUE('
Printer="Nom de l'imprimante"
[;setdefault= True/False ]
[;prnfile="Chemin et Nom du fichier de sortie" ]
[;configfile="Nom du fichier de définition" ]
[;Trunc=True/False ]
[;Source= Nombre ]
[;Orientation= Nombre ]
[;Duplex= Nombre ]
[;Copies= Nombre ]
')
```

```
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Printer = Nom de l'imprimante. Utilisez le nom connu du système Windows.</p> <p>Setdefault = True fixe cette imprimante comme imprimante par défaut.</p> <p>Prnfile = Chemin et Nom du fichier de sortie L'utilisation de cette option est déconseillée en mode CITRIX ou Windows TSE.</p> <p>Configfile = Nom du fichier de définition de l'imprimante Ce fichier est crée par l'utilitaire, voir : Fichiers de configuration</p> |

d'imprimante.**Trunc = True /False**

True : Le fichier désigné par PrnFile est mis à blanc.

False : Les nouvelles impressions seront concaténées au fichier désigné par PrnFile.

Orientation

Portrait = 1

Paysage = 2

Source

Premier = 1

Le plus haut = 1

Un seul = 1

Le plus bas = 2

Milieu = 3

Manuel = 4

Enveloppe = 5

Enveloppe manuelle = 6

Auto = 7

Tracteur = 8

Petit = 9

Large = 10

Large Capacité = 11

Cassette = 14

Source = 15

Dernier = 15

Duplex

1 = Recto simple

2 = Recto verso sur le haut de la page

3 = Recto verso sur le côté gauche

Copies

Nombre de copie à imprimer

Remarques

- Cette commande permet de passer momentanément sur une configuration d'imprimante que vous auriez préenregistrée dans un fichier.



- Lors d'une impression d'un publipostage, d'imprimer dans un fichier.

Ce qui accélère l'exécution des publipostages avec une grande quantité d'enregistrement et permet de contourner certains bugs de Word.

Notamment les problèmes de Word avec des fichiers prenant trop de place en mémoire.

L'impression dans un fichier concatène toujours les nouvelles données avec les données déjà présentes dans le fichier.

L'option Trunc permet de vider le fichier lors de l'appel de la commande SETPRINTER. Si plusieurs impressions sont faites, avant l'appel de RSTPRINTER, les données seront concaténées.

Par exemple, vous pouvez rediriger la sortie de l'imprimante vers un fichier (fichier PRN). En utilisant la commande LAUNCHER IFSPUT et la commande CL LNCPRTDOC, vous pouvez déplacer le fichier PRN vers une OUTQ AS400.

Attention, l'utilisation de l'impression dans un fichier en mode TSE est fortement déconseillée.

Exemple

Cet exemple permet d'imprimer le résultat d'un publipostage dans un fichier texte (.prn), cela correspond dans Word à l'option 'Imprimer dans un fichier'.

//Lancer LAUNCHER

LNCOPEN

MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

//Transférer les données d'un fichier

LNCXFER PCFILE('%LNCDIR%\samples\sp_cust.txt') FROMFILE(SP_CUST)
CLOSECOM(*NO)

MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

//Modifier le fichier de sortie

LNCCMD CMD(SETPRINTER) PARM1('printer="HP LaserJet 1200 Series PCL
6";prnfile="%lnkdir%\samples\test.prn";trunc=true;setdefault=true')

MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

//Ouverture de Microsoft Word Visible

LNCCMD CMD(WORDOPEN) PARM2(visible)

MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

//Fusion du modèle de publipostage vers un fichier texte

LNCCMD CMD(WMAILMERGE) PARM1('document="%lnkdir%\samples\sp_cust.doc"+
;destination=wdSendToPrinter;DataSource="%lnkdir%\samples\sp_cust.txt"
)

MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

//Restaure les paramètres par défaut de l'imprimante

LNCCMD CMD(RSTPRINTER) PARM1('printer="HP LaserJet 1200 Series PCL
6"')

```
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))  
//Fermeture de Microsoft Word  
LNCCMD CMD(WORDCLOSE)
```

Voir aussi

- [Commande RSTPRINTER](#)
- [Commande WMAILMERGE](#)

Commande SHELL

Permet d'exécuter des commandes DOS, d'ouvrir des documents, ou de lancer l'exécution de programmes.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('SHELL')
CHGVAR          VAR(&PARM1) VALUE('commande dos | nom du document')
CHGVAR          VAR(&PARM2) VALUE('
[ ; Wait=True / False / Nombre ]
[ ; Errorlevel=True / False ]
[ ; Visible=True / False ]
[ ; Focus=True / False ]
[ ; Minimize=True / False ]
[ ; OpenDocument=True / False ]
[ ; Directory="Répertoire par défaut" ]
[ ; Action="open" | "explore" | "print" ]
')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Commande DOS à exécuter sur la station Windows, (ex: COPY F:\Docs\Template.doc C:\Temp). Ou, chemin vers un document à ouvrir, ou URL Internet à laquelle se connecter.</p> |
| Parm2 | <p>Wait = True/False/Nombre. Défaut = True. L'application attendra que la commande soit exécutée, sans délai si Wait a la valeur vrai. Si un nombre est exprimé, il indique un délai maximum d'attente en milli secondes.</p> <p>Errorlevel = True / False. Le programme AS/400 recevra le niveau d'erreur en retour de l'appel de la commande. Le niveau d'erreur est retourné dans le paramètre &RESULT, codé sur 5 chiffres.</p> <p>Visible = True / False. Défaut = False. L'exécution de la commande peut se faire en non visible.</p> <p>Focus = True / False. Défaut = True. Si Focus a le valeur faux, la nouvelle application lancée ne prend</p> |

pas le focus Windows.

Dans une application interactive, cela permet d'afficher des données ou images avec une application, et de garder le terminal 5250 toujours actif.

Minimize = True / False. Défaut = False.

Si **Minimize** est vrai, la fenêtre active au moment de l'exécution de cette commande, sera minimisée.

Si **Wait** n'a pas la valeur 'Faux', cette fenêtre sera restaurée à la fin de l'exécution de la commande.

OpenDocument = True / False. Défaut = False.

Quand cette option est vraie, &PARM1 contient le chemin vers un nom de fichier à ouvrir, ou une URL Internet.

Windows utilisera l'application approprié pour le type du fichier désigné.

Directory permet de fixer le répertoire par défaut pour la commande à exécuter, ou le document à ouvrir.

Action indique une action particulière à faire pour un document.

L'option "**OpenDocument**=" est équivalente à **Action**="open".

Action peut prendre les valeurs : "**open**", "**print**", "**explore**".

Remarque

Pour certaines applications, tel que Internet Explorer, il n'est pas possible d'attendre la fin de leur exécution.

Quelle que soit la valeur donnée à l'option **Wait**, la commande est lancée, et votre processus reprend la main immédiatement.

Exemples

Cet exemple copie un fichier d'un répertoire vers un autre.

```
CHGVAR      VAR(&CMD) VALUE('SHELL')
CHGVAR      VAR(&PARM1) VALUE('COPY \Templates\Invoice.doc
C:\Temp\Invoice.doc')
CHGVAR      VAR(&PARM2) VALUE('ErrorLevel')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Cet exemple connecte la station Windows à un site Web.

```
CHGVAR      VAR(&CMD) VALUE('SHELL')
CHGVAR      VAR(&PARM1) VALUE('http://www.easycom-aura.com')
```



```
CHGVAR      VAR(&PARM2) VALUE('OpenDocument')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Cet exemple ouvre une page HTML avec le navigateur par défaut.

```
CHGVAR      VAR(&CMD) VALUE('SHELL')
CHGVAR      VAR(&PARM1) VALUE('\Pages\Document.htm')
CHGVAR      VAR(&PARM2) VALUE('OpenDocument')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande *STO*

Mémorise la fenêtre active, et retourne son numéro.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('STO')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Le numéro de la fenêtre active est retourné sur 10 chiffres, dans le paramètre &RESULT.

Cette fenêtre pourra être restaurée par la commande RCL.

Voir aussi

- [MIN](#)
- [RCL](#)

Commande VAL

Permet d'entrer une valeur dans une des cellules du fichier de fusion, créé par la commande CRF.

L'entrée de la valeur se fera toujours dans la ligne courante.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('VAL')
CHGVAR          VAR(&PARM1) VALUE('nom_colonne')
CHGVAR          VAR(&PARM2) VALUE('valeur')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Nom de la colonne ou numéro de la colonne (exemple : #2). |
| Parm2 | Valeur à entrer. |

Exemple

```
LNCCMD          CMD(CRF) PARM1('C:\A\fusion2.txt')
LNCCMD          CMD(NCL) PARM1('Nom')
LNCCMD          CMD(NCL) PARM1('Prenom')
LNCCMD          CMD(NCL) PARM1('Identifiant')
LNCCMD          CMD(NLN)
LNCCMD          CMD(VAL) PARM1('Nom') PARM2('Redfall')
LNCCMD          CMD(VAL) PARM1('Prenom') PARM2('Robert')
LNCCMD          CMD(VAL) PARM1('Identifiant') PARM2('1')
LNCCMD          CMD(ENF)
LNCCLOSE
```

Dans cet exemple, le fichier 'fusion2.txt' est créé dans le répertoire "C:\A".

Voir aussi

- [CRF](#)
- [NLN](#)
- [NCL](#)
- [ENF](#)

Commande WADDINS

Permet de charger un complément Microsoft Word.

Un complément est un programme supplémentaire qui ajoute des commandes ou des fonctionnalités personnalisées à Microsoft Word.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WADDINS')
CHGVAR      VAR (&PARM1) VALUE ('Complément')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin complet vers le complément à charger. |
| Parm2 | |

Remarque

On peut utiliser WADDINS pour ajouter un modèle qui contient des macros et les rendre ainsi disponibles.

Exemple

Chargement du complément fourni avec Launcher pour créer un menu personnalisé avec Launcher (commande WMENU) :

```
CHGVAR      VAR (&CMD) VALUE ('WADDINS')
CHGVAR      VAR (&PM1) VALUE ('%LNCDIR%\LNCWordAddin.dot')
CHGVAR      VAR (&PM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HDL &CMD &OPT &PM1 &PM2 &RES)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Autre exemple

Affecter un style dans un document Word par macro :

```
PGM
LNCOOPEN
LNCCMD CMD(WORDOPEN)
```

```
LNCCMD CMD (WOPENFILE) +  
PARM1 ('%LNCDIR%\SAMPLES\MODELE_STYLE.DOC')  
LNCCMD CMD (WADDINS) PARM1 ('%LNCDIR%\SAMPLES\MODELE_MACRO.dot')  
LNCCMD CMD (WMENU) +  
PARM1 ('Captions="macro2";Tips="macro2"')  
LNCCMD CMD (WORDWAIT)  
LNCCMD CMD (WORDSHOW)  
LNCCLOSE  
ENDPGM
```

Voir aussi

- [WMENU](#)
- [WORDWAIT](#)
- [WGETPROP](#)
- [WEXEMACRO](#)

Commande *WBOOKMADD*

Ajoute un signet à la position courante du curseur.

Le nom du signet ajouté doit respecter les contraintes imposées par Word. La sélection courante est identifiée par le nom du signet ajouté, et le programme pourra se repositionner à cette même position par la commande WBOOKMARK.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WBOOKMADD')
CHGVAR      VAR(&PARM1) VALUE('
Bookmark="Nom du bookmark à insérer"]
[;Text="Texte à insérer" ]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Bookmark : Nom du signet à insérer dans le document courant. |
| | Text : Valeur du texte à insérer à l'emplacement du signet créé. Optionnel. |

Exemple

```
DCL        VAR(&FILE) TYPE(*CHAR) LEN(2000)
DCL        VAR(&PATH) TYPE(*CHAR) LEN(2000)

CHGVAR     VAR(&PATH) VALUE('C:\A\')
CHGVAR     VAR(&FILE) VALUE('res.docx')

LNCOPEN    SVRADDR('*DEV')

LNCCMD     CMD(WORDOPEN)

LNCCMD     CMD(WOPENFILE) PARM1(&PATH *TCAT &FILE)

LNCCMD     CMD(WBOOKMADD) +
           PARM1('Bookmark="Name";Text="Client name"')

LNCCMD     CMD(WSAVEAS) +
           PARM1('FILE="C:\A\testBookmarkAdd.docx"')

LNCCMD     CMD(WORDCLOSE)
```

LNCCLOSE

Voir aussi

- [WBOOKMARK](#)



Commande WBOOKMARK

Insère du texte à l'emplacement du signet spécifié.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE (WBOOKMARK)
CHGVAR          VAR(&PARM1) VALUE ('
                [ Bookmark="Nom de signet" ] +
                [ ;Start]
                [ ;End]
                [ ;Overtime]
                [ ;Adjust]
                [ ;Append] +
                [ ;InsertRow]
                [ ;Merge=n] +
                [ ;Unicode=True/False ] +
                [ ;ToBookmark="Nom de signet" ]
                [ ;NUMFMT(d DECPOINT=p GRPPOINT=g) ] +
                [ ;PROP (Property)=Value] +
                ')

CHGVAR          VAR(&PARM2) VALUE ('Texte à insérer')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Bookmark est le nom d'un signet présent dans le document, ou un signet prédéfini de Microsoft Word, ou une valeur spéciale - Voir les notes plus bas.</p> <p>Un signet représente une position dans le document, ou une sélection (plage) de plusieurs caractères. Un signet se place dans un document Word par le menu « Insertion » « Signet » de Word.</p> <p>Si le nom de signet est absent dans Parm1, ou, s'il est égal à *NONE, alors, le curseur Word n'est pas déplacé, et le texte sera inséré à la position courante.</p> <p>Si le signet spécifié ici est absent du document, aucune action n'est effectuée, et le mot 'FALSE' est retourné dans le paramètre &RESULT.</p> <p>Start : Quand le signet englobe une sélection, l'option Start indique que l'on veut positionner le curseur de Word au début de la sélection.</p> <p>End : Quand le signet englobe une sélection, l'option End indique</p> |

que l'on veut positionner le curseur de Word à la fin de la sélection.

Overtyp indique que le nouveau texte vient écraser le texte existant. Par défaut, le nouveau texte sera inséré, ou, remplacera la sélection.

Adjust permet de replacer le signet à l'emplacement du texte inséré.

Append ajoute la nouvelle valeur au contenu actuel du signet.

InsertRow permet d'insérer une ligne dans un tableau, à l'emplacement du signet.

Merge : fusionne les n cellules en une seule.

NUMFMT(d DECPOINT=p GRPPOINT=g) permet de formater une valeur numérique décimale.

-**d** est le nombre de décimales.

-**p** représente le caractère à utiliser comme point décimal.

-**g** représente le caractère à utiliser comme séparateur de milliers.

PROP fixe la valeur d'une propriété à l'emplacement du texte inséré.

Plusieurs directives **PROP()** peuvent se suivre, séparées par un point-virgule (;).

Exemple :

```
PROP(Font.bold)=True;Prop(Font.italic)=True
```

Unicode : Par défaut Unicode=false. Quand Unicode est vrai, le texte à écrire dans &PARM2 doit être dans le jeu de caractère Unicode de Windows.

La valeur Unicode Windows peut être générée par le programme [LNCCVTWCS](#).

ToBookmark permet de désigner un signet. La sélection sera étendue jusqu'à ce signet, à partir du signet spécifié avec le paramètre Bookmark, ou de la position courante si Bookmark n'est pas spécifié.

Parm2

Texte à insérer.

Si le texte est absent (blanc), le curseur est positionné sur le signet, et aucun texte n'est inséré.

La valeur spéciale **%NONE%** entraîne l'effacement de la valeur actuelle de la sélection.

Si l'option **Unicode=** est fixée, le contenu de &Parm2 doit être en format Unicode Windows, et le paramètre **&OPT** doit avoir la valeur **'W'**.

| | |
|-----|---|
| Opt | Quand Unicode est vrai, &OPT doit avoir la valeur 'W' . &PARM2 doit contenir une chaîne Unicode Windows. |
|-----|---|

Remarques

1) Le nom du signet peut prendre les valeurs suivantes :

- Un nom de signet inséré dans le modèle du document
- Un [Signet Prédéfini](#) de Microsoft Word
- ***RIGHT** pour se déplacer vers la cellule de droite
- ***LEFT** pour se déplacer vers la cellule de gauche
- ***UP** pour se déplacer à la cellule du dessus
- ***DOWN** pour se déplacer à la cellule du dessous

Si le signet n'existe pas, aucun texte n'est inséré.

2) Si le signet englobe une sélection de plusieurs caractères, c'est toute la sélection qui sera remplacé par la nouvelle valeur.

3) La commande WBOOKMARK peut servir à supprimer des paragraphes entiers qui sont compris dans un signet.

4) Les signets peuvent se chevaucher.

5) Les signets peuvent se trouver n'importe où dans le document, y compris dans les entêtes et pieds de page, ou les filigranes.

6) Word adapte le mode d'affichage automatiquement.

7) Dans tous les cas, le déplacement s'effectue avant l'insertion de texte.

Exemple 1

```
CHGVAR VAR(&CMD) VALUE('WBOOKMARK')
CHGVAR VAR(&PARM1) VALUE('Bookmark="Signet1";NUMFMT(0)')
CHGVAR VAR(&PARM2) VALUE('0001234')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Résultat : 1234

```
CHGVAR VAR(&CMD) VALUE('WBOOKMARK')
```



```
CHGVAR VAR(&PARM1) VALUE('Bookmark="Signet2";NUMFMT(2)')
CHGVAR VAR(&PARM2) VALUE('0001234')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Résultat : 12.34

Exemple 2

```
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('FILE ="C:\TEMP\TEST.DOCX"')
CHGVAR VAR(&PARM2) VALUE('VISIBLE')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR VAR(&CMD) VALUE('WBOOKMADD')
CHGVAR VAR(&PARM1) VALUE('Bookmark="S1"')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR VAR(&CMD) VALUE('WBOOKMARK')
CHGVAR VAR(&PARM1) VALUE('Bookmark="S1";NUMFMT(2 DECPOINT=,
GRPPOINT=.)')
CHGVAR VAR(&PARM2) VALUE('12345678')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Exemple 3

```
LNCCMD      CMD(WBOOKMADD) +
            PARM1('Bookmark="Name";Text="Client name"')
```

```
LNCCMD      CMD(WBOOKMARK) +
            PARM1('Bookmark="Name";Start;Overtyp;Adjust;
PROP(Font.bold)=True;Prop(Font.italic)=True')
            PARM2('Aura Equipements')
```

Exemple RPG

La valeur 12345678 envoyée par l'AS/400 avec le format suivant :

NUMFMT(2 DECPOINT=, GRPPOINT =.)

Imprime la valeur ainsi:

123.456,78

L'exemple RPG suivant :

```
EVAL  LNC_PARM2  =*BLANKS
EVAL  LNC_PARM1=' *RIGHT NUMFMT (2) '
MOVE  TOTAL  LNC_PARM2
CALL  'LNCCMD'
```

. . .

va permettre d'avoir une donnée numérique du côté AS/400 et une zone texte formatée sur le document.

Si la variable *TOTAL* vaut **12350** (la donnée envoyée à LAUNCHER Office est représentée comme celle que vous pouvez voir dans UPDDTA, sans le point décimal) dont deux chiffres pour la partie décimale. LAUNCHER Office placera dans le document la valeur **123.50**.

Comment formater une date ?

Il n'existe pas de commandes LAUNCHER Office pour WORD permettant de formater des dates.

L'utilisateur doit donc utiliser la commande "CVTDAT" sur l'AS/400.

Cependant, dans le cas d'un publipostage, il est possible de définir le format des dates dans le document WORD pour tous les champs "MERGEFIELD".

Voir aussi

- [Les valeurs spéciales](#)
- [LNCWBM - Programme API](#)

Commande *WBOOKMDEL*

Supprime un signet ou tous les signets du document.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WBOOKMDEL')
CHGVAR          VAR(&PARM1) VALUE('Bookmark="Nom du signet"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Bookmark : nom du signet à supprimer dans le document courant, ou "*ALL" pour supprimer tous les signets du document, ou "Pref*" pour supprimer les signets dont le nom commence par "Pref". |

Remarque

Cette commande est utilisée lorsqu'une application doit insérer dans le document courant, un autre modèle de document, ou le même modèle, contenant les mêmes noms de signets.

Si les noms de signets du document inséré existent déjà dans le document courant, alors Word conserve les signets existants.

Exemple

```
CHGVAR          VAR(&CMD) VALUE('WBOOKMDEL')
CHGVAR          VAR(&PARM1) VALUE('*ALL')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

CHGVAR          VAR(&CMD) VALUE('WINSERTF')
CHGVAR          VAR(&PARM1) VALUE('File="%MASTERS%\MyDocument.docx" +
Bookmark ="\EndOfDoc"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```



Commande WCHDIR

Change le répertoire courant de Word.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WCHDIR')
CHGVAR          VAR(&PARM1) VALUE('Directory="Chemin de répertoire"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Directory : Indique le chemin de répertoire à partir duquel Word recherchera les documents ou autres fichiers, quand le chemin complet ne sera pas spécifié. Ce chemin de recherche est aussi utilisé par certaines directives Word : IncludeText, IncludePicture, ... |

Exemple

```
CHGVAR          VAR(&CMD) VALUE('WCHDIR')
CHGVAR          VAR(&PARM1) VALUE('Directory ="%TEMP%")
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande *WCLOSEFILE*

Ferme le document **WORD** courant.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WCLOSEFILE')
CHGVAR      VAR (&PARM1) VALUE (' [*YES | *NO] ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Option de sauvegarde *YES : sauvegarde des modifications puis fermeture du document. *NO : ferme le document sans sauvegarder les modifications apportées. Par défaut : Si ce document n'a pas été sauvegardé auparavant, et si aucune option n'est choisie, une boîte de dialogue apparaîtra pour demander le nom de sauvegarde. |

Exemple

```
CHGVAR      VAR (&CMD) VALUE ('WCLOSEFILE')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

ou

```
LNCCMD      CMD (WCLOSEFILE)
```

Dans le dernier exemple, **WORD** affichera une boîte de dialogue demandant la sauvegarde. Si **WORD** est réduit, il clignotera dans la barre des tâches. Restaurez **WORD** pour faire apparaître la boîte de dialogue.

Voir aussi

- [WOPENFILE](#)
- [WSAVEAS](#)

Commande WCOPY

Copie la sélection courante ou le contenu d'un signet, dans le presse-papiers ou dans un autre signet.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WCOPY')
CHGVAR          VAR(&PARM1) VALUE('
[From="Signet source"];
[To="Signet de destination"];
[Count=Nombre]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>From : Désigne un signet dans le document, dont le contenu doit être copié. Si From n'est pas précisé, c'est la sélection courante qui est copiée vers le presse papier.</p> <p>To : Désigne un signet dans le document, vers lequel doit se faire la copie. Si To n'est pas précisé, les données sont copiées dans le presse papier uniquement.</p> <p>Count : Indique le nombre de copies à faire. Ce paramètre n'est valide que si To est précisé.</p> |

Exemples

L'exemple suivant copie la valeur 123 456,78 de "Ligne1" vers le presse papier

```
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('FILE ="C:\TEMP\TEST.DOC"')
CHGVAR VAR(&PARM2) VALUE('VISIBLE')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR VAR(&CMD) VALUE('WBOOKMADD')
```



```

CHGVAR VAR(&PARM1) VALUE('Ligne1')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

CHGVAR VAR(&CMD) VALUE('WBOOKMARK')
CHGVAR VAR(&PARM1) VALUE('Ligne1;NUMFMT(2 DECPOINT=, GRPPOINT=.)')
CHGVAR VAR(&PARM2) VALUE('12345678')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

CHGVAR VAR(&CMD) VALUE('WSELECT')
CHGVAR VAR(&PARM1) VALUE('*LINE')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

CHGVAR VAR(&CMD) VALUE('WCOPY')
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

```

L'exemple suivant copie 10 fois le contenu de "Ligne1" à l'emplacement de "Ligne2".

```

CHGVAR          VAR(&CMD) VALUE('WCOPY')
CHGVAR          VAR(&PARM1) VALUE('From ="Ligne1";To="Ligne2";Count=10')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG          MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

```

L'exemple suivant copie la sélection courante vers le presse papier.

```

CHGVAR          VAR(&CMD) VALUE('WCOPY')
CHGVAR          VAR(&PARM1) VALUE(' ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG          MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

```

Voir aussi

- [WPASTE](#)



- [WSELECT](#)



Commande WDOCUMENT

Ouvre document WORD.

Cette commande synthétique revient à faire un [WOPENFILE](#) ou [WNEWFILE](#) avec la possibilité de lui passer en paramètre l'équivalent des commandes [WORDSHOW/WORDHIDE](#) et [WMAXIMIZE/WMINIMIZE](#).

Il faut avoir ouvert Word au préalable par la commande [WORDOPEN](#).

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WDOCUMENT')
CHGVAR      VAR(&PARM1) VALUE('Nom Fichier' OU ' ' 'Nom Modèle')
CHGVAR      VAR(&PARM2) VALUE('
[Directory="Chemin de répertoire";]
[Document="Nom de document";]
[Detach = True/False;]
[Visible;]
[ReadOnly;]
[New;]
[Minimize;]
[Maximize;]
[DOCPWD="Mot de passe Ouverture";]
[WRTPWD="Mot de passe Modifications";]
')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Nom du fichier à ouvrir, peut être vide pour la création d'un nouveau document ou contenir le nom d'un modèle si on spécifie NEW en paramètre 2 |
| Parm2 | <p>Directory Cette option permet de spécifier un chemin vers le répertoire où se trouve le document à ouvrir. Ce chemin de répertoire est ajouté devant le chemin et le nom de fichier donné en paramètre 1.</p> <p>Document Par le mot clé "Document", on demande à LAUNCHER Office de rechercher le document parmi ceux déjà ouverts par Word sur le système. Si le document demandé est déjà ouvert, il est rendu actif, il devient le document courant pour Word. Aucun nouveau fichier est ouvert, le nom du fichier ou du modèle sont alors ignorés.</p> <p>Detach = True/False (false par défaut) Si "Detach" est vrai, Word détache la source de données du</p> |



document courant ou du nouveau document actif au retour de l'appel de la commande WDOCUMENT. En détachant la source de données, le document n'est plus un modèle de publipostage pour Word. La commande WMAILMERGE refera l'attachement à une source de données.

Quand un document est ouvert et attaché à une source de données (Fichier de fusion), il est tenu par Word, et donc non modifiable. Il faut détacher la source de données du document courant si on veut envoyer des nouvelles données vers le fichier de fusion.

VISIBLE : pour rendre Word visible.

MINIMIZE : fenêtre réduite.

MAXIMIZE : fenêtre agrandie.

READONLY : ouvrir en lecture seule.

NEW : si Parm1 est vide, permet la création d'un nouveau document.

DOCPWD : mot de passe pour l'ouverture. Si le document est protégé par un mot de passe, il faut respecter la casse.

WRTPWD : mot de passe pour les modifications. Si le document est protégé contre les modifications, renseigner le mot de passe.

Attention !!! Word doit être visible pour ouvrir un document protégé par mot de passe.

RESULT

Nom du document

Au retour de la commande WDOCUMENT, le paramètre &RESULT contient le nom du document actif.

Pour un document ouvert depuis un fichier, le nom du document est égal au nom du fichier, sans le chemin et sans le suffixe ".DOCX". Si le document vient d'être créé, son nom a été construit par Word.

Exemple

Crée un nouveau document dans une fenêtre agrandie et visible.

```
CHGVAR          VAR ( &CMD)  VALUE ( 'WDOCUMENT ' )
```



```
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE('VISIBLE;MAXIMIZE;NEW')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Ouvre le document Rapport.docx en lecture seule dans une fenêtre réduite

```
CHGVAR      VAR(&CMD) VALUE('WDOCUMENT')
CHGVAR      VAR(&PARM1) VALUE('C:\temp\Rapport.docx')
CHGVAR      VAR(&PARM2) VALUE('VISIBLE;MINIMIZE;READONLY')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WOPENFILE](#)
- [WNEWFILE](#)

Commande *WDPROTECT*

Permet d'enlever la protection sur un document.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WDPROTECT')
CHGVAR      VAR(&PARM1) VALUE('MotDePasse')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Le mot de passe spécifié à l'appel de la commande WPROTECT. |
| Parm2 | |

Exemple

Supposons qu'un document Word ait été protégé par WPROTECT, de cette façon :

```
LNCCMD      CMD(WORDOPEN)
LNCCMD      CMD(WOPENFILE) PARM1('c:\A\resProtect.docx')
LNCCMD      CMD(WPROTECT) +
             PARM1('Level=wdAllowOnlyComments;PWD="Aura"')
LNCCMD      CMD(WSAVE)
LNCCMD      CMD(WORDCLOSE) PARM1('SAVE=FALSE')
```

Pour enlever la protection, il faudra utiliser la commande suivante :

```
LNCCMD      CMD(WDPROTECT) PARM1('Aura')
```

Voir aussi

- [WPROTECT](#)

Commande WEXEMACRO

Exécute une macro Microsoft Word.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WEXEMACRO')
CHGVAR      VAR(&PARM1) VALUE('Nom Macro')
CHGVAR      VAR(&PARM2) VALUE('[Paramètres de la macro]')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Nom de la macro. Parfois le nom complet de la macro est nécessaire. Exemple : Modèle.Module.Macro. |
| Parm2 | Paramètres passés à la macro. Les paramètres doivent être entre guillemets, séparés par un point virgule. |

Exemple 1

```
CHGVAR      VAR(&CMD) VALUE('WEXEMACRO')
CHGVAR      VAR(&PARM1) VALUE('Macro1')
CHGVAR      VAR(&PARM2) VALUE('"12000";"FRANCE" ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Exemple 2

Supposons que nous ayons la macro suivante avec 2 paramètres :

```
Sub InsertTextAtEndOfDocument(param1 as string, param2 as string)
ActiveDocument.Content.InsertAfter Text:= param1
ActiveDocument.Content.InsertAfter Text:= param2
End Sub
```

Pour exécuter la macro, on peut lancer la commande suivante :

```
LNCCMD      CMD(WEXEMACRO) +
            PARM1('InsertTextAtEndOfDocument') +
            PARM2('"Aura";"Equipements"')
```

Voir aussi

- [WADDINS](#)
- [WGETPROP](#)

Commande *WFIELDS*

Insère un champ dans un document Microsoft Word.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WFIELDS')
CHGVAR      VAR(&PARM1) VALUE('
Text="Expression à insérer"
[;Type=Valeur]
[;Bookmark="Signet"]
[;PreserveFormat=True/False]
[;Update=True/False]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Text = Texte additionnel pour la création du champ.</p> <p>Type = Type de champ à ajouter. (Défaut = <i>wdFieldTypeEmpty</i>). Voir les valeurs acceptées plus bas.</p> <p>Bookmark = désigne un signet existant dans le document, où l'insertion du nouveau champ doit se faire.</p> <p>PreserveFormat = permet de conserver le format du champ lors de chaque mise à jour.</p> <p>Update = permet de mettre à jour tous les champs du document.</p> |

Remarque

Valeurs acceptées pour le type de champ :

| | | |
|------------------------------|------------------------|----------------------------|
| <i>wdFieldAddin</i> | <i>wdFieldAdvance</i> | <i>wdFieldAsk</i> |
| <i>wdFieldAuthor</i> | <i>wdFieldAutoNum</i> | <i>wdFieldAutoNumLegal</i> |
| <i>wdFieldAutoNumOutline</i> | <i>wdFieldAutoText</i> | <i>wdFieldAutoTextList</i> |
| <i>wdFieldBarCode</i> | <i>wdFieldComments</i> | <i>wdFieldCompare</i> |

| | | |
|------------------------------|----------------------------|----------------------------|
| <i>wdFieldCreateDate</i> | <i>wdFieldData</i> | <i>wdFieldDatabase</i> |
| <i>wdFieldDate</i> | <i>wdFieldDDE</i> | <i>wdFieldDDEAuto</i> |
| <i>wdFieldDocProperty</i> | <i>wdFieldDocVariable</i> | <i>wdFieldEditTime</i> |
| <i>wdFieldEmbed</i> | <i>wdFieldEmpty</i> | <i>wdFieldExpression</i> |
| <i>wdFieldFileName</i> | <i>wdFieldFileSize</i> | <i>wdFieldFillIn</i> |
| <i>wdFieldFootnoteRef</i> | <i>wdFieldFormCheckBox</i> | <i>wdFieldFormDropDown</i> |
| <i>wdFieldFormTextInput</i> | <i>wdFieldFormula</i> | <i>wdFieldGlossary</i> |
| <i>wdFieldGotoButton</i> | <i>wdFieldHTMLActiveX</i> | <i>wdFieldHyperlink</i> |
| <i>wdFieldIf</i> | <i>wdFieldImport</i> | <i>wdFieldInclude</i> |
| <i>wdFieldIncludePicture</i> | <i>wdFieldIncludeText</i> | <i>wdFieldIndex</i> |
| <i>wdFieldIndexEntry</i> | <i>wdFieldInfo</i> | <i>wdFieldKeyWord</i> |
| <i>wdFieldLastSavedBy</i> | <i>wdFieldLink</i> | <i>wdFieldListNum</i> |
| <i>wdFieldMacroButton</i> | <i>wdFieldMergeField</i> | <i>wdFieldMergeRec</i> |
| <i>wdFieldMergeSeq</i> | <i>wdFieldNext</i> | <i>wdFieldNextIf</i> |
| <i>wdFieldNoteRef</i> | <i>wdFieldNumChars</i> | <i>wdFieldNumPages</i> |
| <i>wdFieldNumWords</i> | <i>wdFieldOCX</i> | <i>wdFieldPage</i> |
| <i>wdFieldPageRef</i> | <i>wdFieldPrint</i> | <i>wdFieldPrintDate</i> |
| <i>wdFieldPrivate</i> | <i>wdFieldQuote</i> | <i>wdFieldRef</i> |
| <i>wdFieldRefDoc</i> | <i>wdFieldRevisionNum</i> | <i>wdFieldSaveDate</i> |
| <i>wdFieldSection</i> | <i>wdFieldSectionPages</i> | <i>wdFieldSequence</i> |
| <i>wdFieldSet</i> | <i>wdFieldSkipIf</i> | <i>wdFieldStyleRef</i> |
| <i>wdFieldSubject</i> | <i>wdFieldSubscriber</i> | <i>wdFieldSymbol</i> |
| <i>wdFieldTemplate</i> | <i>wdFieldTime</i> | <i>wdFieldTitle</i> |
| <i>wdFieldTOA</i> | <i>wdFieldTOAEntry</i> | <i>wdFieldTOC</i> |
| <i>wdFieldTOCEntry</i> | <i>wdFieldUserAddress</i> | <i>wdFieldUserInitials</i> |
| <i>wdFieldUserName</i> | | |

Exemples

Cet exemple ajoute le champ variable Word "Variable1" et lui affecte la valeur "Monsieur".

Les guillemets sont doublés autour de la valeur.

```
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('FILE="C:\TEMP\TEST.DOCX"')
CHGVAR VAR(&PARM2) VALUE('VISIBLE')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR VAR(&CMD) VALUE('WFIELDS')
CHGVAR VAR(&PARM1) VALUE('TEXT="SET variable1 ""MONSIEUR""";+
```

```
TYPE= WDFIELDDOCVARIABLE;BOOKMARK="\STARTOFDOC")
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

L' exemple suivant met à jour tous les champs du document.

```
CHGVAR      VAR(&CMD) VALUE('WFIELDS')
CHGVAR      VAR(&PARM1) VALUE('Update =True')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Enfin cet exemple crée un bookmark et assigne un champ de fusion nommé CUST_ID sur ce bookmark :

```
LNCCMD      CMD(WBOOKMADD) +
             PARM1('BOOKMARK="Name";TEXT="Aura"')

LNCCMD      CMD(WFIELDS) +
             PARM1('Text="CUST_ID";Type=wdFieldMergeField;Bookmark="Name"')
```

Commande *WFINDTEXT*

Permet de rechercher du texte dans le document courant.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WFINDTEXT')
CHGVAR      VAR(&PARM1) VALUE('Text="Texte à rechercher"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Text = Texte à rechercher entre guillemets. |
| RESULT | Si le texte est trouvé, la variable &RESULT renvoie *TRUE . Sinon, elle renvoie *FALSE . |

Exemple

```
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('FILE ="C:\TEMP\TEST.DOCX"')
CHGVAR VAR(&PARM2) VALUE('VISIBLE')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR VAR(&CMD) VALUE('WFINDTEXT')
CHGVAR VAR(&PARM1) VALUE('Text="Aura"')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Autre exemple

```
LNCCMD CMD(WFINDTEXT) PARM1('Text="1234"')
```

Commande *WFORMFIELD*

Permet de modifier par programme la valeur d'un champ d'un formulaire Microsoft Word.

Le document Word doit être verrouillé (Formulaire) pour pouvoir modifier les valeurs des champs.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WFORMFIELD')
CHGVAR          VAR(&PARM1) VALUE('
Field="Nom Signet"
[;NUMFMT(d DECPOINT=p GRPPOINT=g)]
[;Clear=True/False]
[;AddEntries=True/False]
[;Value=Indice_de_liste]
[;SetText=True/False]
[;GetValue=True/False]
[;GetText=True/False]
')
CHGVAR          VAR(&PARM2) VALUE('[texte]')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Field = nom du signet attribué au champ dans le document courant.</p> <p>NUMFMT(d DECPOINT=p GRPPOINT= permet de formater une valeur numérique décimale.</p> <ul style="list-style-type: none"> -d est le nombre de décimales. -p représente le caractère à utiliser comme point décimal. -g représente le caractère à utiliser comme séparateur de milliers. <p>Clear permet d'effacer le contenu du champ avant de lui affecter de nouvelles valeurs. Si le champ est une liste déroulante, toutes les entrées sont supprimées.</p> <p>AddEntries permet d'ajouter des entrées dans une liste déroulante. Le paramètre 2 (&PARM2) contient alors la liste des valeurs, entre guillemets, et séparées par "point virgule".</p> <p>Value permet de changer l'indice de la valeur en cours dans une liste déroulante, ou dans une boîte à cocher.</p> |

Si **SetText** est vrai, alors le contenu du paramètre 2 sera considéré comme un texte à affecter au champ, et non comme l'indice de la valeur en cours dans une liste.

Si **SetText** est faux, alors le contenu du paramètre 2 sera considéré comme un indice, s'il est numérique.

Quand **GetValue** est vrai, l'indice en cours d'une liste déroulante est retourné dans le paramètre &RESULT.

Quand **GetText** est vrai, la valeur du champ est retourné dans le paramètre &RESULT.

Parm2 Texte à insérer, ou liste des entrées à ajouter à une liste déroulante.

Exemple

L'exemple suivant donne au champ "FF01" la valeur de la variable **&FIELDVAL**.

```
CHGVAR            VAR(&CMD) VALUE('WFORMFIELD)
CHGVAR            VAR(&PARM1) VALUE('Field ="FF01";SetText')
CHGVAR            VAR(&PARM2) VALUE(&FIELDVAL)
CALL              PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                  &PARM2 &RESULT))
```

L'exemple suivant fixe les entrées d'une liste déroulante.

```
CHGVAR            VAR(&CMD) VALUE('WFORMFIELD)
CHGVAR            VAR(&PARM1) VALUE('Field ="FF02";AddEntries')
CHGVAR            VAR(&PARM2) VALUE('"France";"Italie";"Espagne"')
CALL              PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                  &PARM2 &RESULT))
```

L'exemple récupère la valeur texte d'un champ de formulaire, puis l'efface.

```
CHGVAR            VAR(&CMD) VALUE('WFORMFIELD)
CHGVAR            VAR(&PARM1) VALUE('Field= ="FF03";GetText;Clear')
CHGVAR            VAR(&PARM2) VALUE(' ')
CALL              PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                  &PARM2 &RESULT))
```

Commande WGETHPOS

Récupère la position horizontale du curseur depuis le bord gauche de la page.

C'est la distance entre le curseur et le bord gauche de la page mesurée en points (1 point = 20 twips, 72 points = 1 pouce, 1 pouce = 2.54cm).

Le résultat est envoyé via la variable &RESULT.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WGETHPOS')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Commande *WGETPAGE*

Récupère le numéro de la page active du document Microsoft Word en cours d'utilisation.

Le résultat est récupéré via la variable &RESULT.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WGETPAGE')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Exemple

```
CHGVAR VAR (&CMD) VALUE ('WORDOPEN')
CHGVAR VAR (&PARM1) VALUE ('C:\TEMP\FORMULAIRE.DOCX')
CHGVAR VAR (&PARM2) VALUE ('VISIBLE')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR VAR (&CMD) VALUE ('WGETPAGE')
CHGVAR VAR (&PARM1) VALUE (' ')
CHGVAR VAR (&PARM2) VALUE (' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande WGETPROP

Permet de lire la valeur d'une propriété de Word.

Word et Excel utilisent le même modèle d'objets de Visual Basic Application. On peut visualiser l'ensemble des objets et leurs propriétés à partir de l'explorateur d'objet de Visual Basic (Outils – Macro – Visual Basic Editor ou Alt-F11) puis Affichage – Explorateur d'Objets (ou F2).

Syntaxe 1

```
CHGVAR      VAR(&CMD) VALUE('WGETPROP')
CHGVAR      VAR(&PARM1) VALUE('chemin propriété')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Chemin complet de la propriété Word.</p> <p>La syntaxe du chemin vers la propriété respecte les mêmes règles que pour XLGETPROP.</p> <p>La valeur de la propriété est retournée sous forme de chaîne de caractères dans la variable &RESULT.</p> |
| RESULT | Valeur de la propriété spécifiée dans Parm1. |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WGETPROP')
CHGVAR      VAR(&PARM1) VALUE('Selection.Font.Color')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Syntaxe 2

Utilisation du paramètre **Property**. Nécessaire pour l'appel des webservices.

```
CHGVAR      VAR(&CMD) VALUE('WGETPROP')
CHGVAR      VAR(&PARM1) VALUE('Property="Chemin propriété"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
|------------|--|

| | |
|--------|---|
| Parm1 | Property = Chemin complet de la propriété Word. La syntaxe du chemin vers la propriété respecte les mêmes règles que pour XLGETPROP . La valeur de la propriété est retournée sous forme de chaîne de caractères dans la variable &RESULT. |
| Result | Valeur de la propriété spécifiée avec le paramètre Property . |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WGETPROP')
CHGVAR      VAR(&PARM1) VALUE('Property="Selection.Font.Color"')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [WSETPROP](#)

Commande WGETTEXT

Permet de récupérer le contenu :

- de la sélection en cours
- d'un signet
- d'un champ de formulaire

du document Word en cours d'utilisation.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WGETTEXT')
CHGVAR          VAR(&PARM1) VALUE('
                [Bookmark="Nom de signet"]
                [;Variable="Nom de variable"]
                [;Field="Nom d'un champ"]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Bookmark : nom du signet dont on veut récupérer la valeur.</p> <p>Variable : nom de la variable dont on veut récupérer la valeur.</p> <p>Field : nom du champ de formulaire dont on veut récupérer la valeur.</p> <p>Si Parm1 est vide, c'est la sélection en cours qui est récupérée.</p> |
| RESULT | Contenu de l'élément spécifié. |

Exemple

Récupère le contenu d'un signet :

```

CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('C:\TEMP\FORMULAIRE.DOCX')
    
```

```
CHGVAR VAR(&PARM2) VALUE('VISIBLE')  
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +  
&PARM2 &RESULT)
```

```
CHGVAR VAR(&CMD) VALUE('WGETTEXT')  
CHGVAR VAR(&PARM1) VALUE('Bookmark="FF03"')  
CHGVAR VAR(&PARM2) VALUE(' ')  
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +  
&PARM2 &RESULT)
```

Voir aussi

- [WSELECT](#)

Commande *WGETVPOS*

Récupère la position verticale du curseur depuis le haut de la page.

C'est la distance entre le curseur et le bord supérieur de la page mesurée en points (1 point = 20 twips, 72 points = 1 pouce, 1 pouce = 2.54cm).

Le résultat est envoyé via la variable &RESULT.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WGETVPOS')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Commande WGOTOTBL

Déplace le curseur sur le prochain tableau dans le document Word actif.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WGOTOTBL')
CHGVAR          VAR (&PARAM1) VALUE (' ')
CHGVAR          VAR (&PARAM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
MONMSG         MSGID (LNC0000) EXEC (GOTO CMDLBL (ERROR))
```

Commande *WINSERTOBJ*

Insère un objet OLE dans le document Word actif.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WINSERTOBJ')
CHGVAR      VAR(&PARM1) VALUE('
File="Chemin vers l'objet";
[;Bookmark="Signet"]
[;Link=True/False]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>File : Chemin complet vers le fichier objet à insérer.</p> <p>Bookmark :(Optionnel) Nom du signet à l'emplacement duquel le nouveau document doit être inséré. Les signets prédéfinis de Word sont utilisables.</p> <p>Link : True/False, False par défaut. Si Link est vrai, c'est un lien vers l'objet qui est inséré. Les modifications apportées à l'objet seront automatiquement reportées sur les documents avec lequel il est lié.</p> |

Exemple

Insertion de slides Powerpoint dans un document Word :

```
LNCCMD CMD(WINSERTOBJ) PARM1('File="C:\temp\stored_proc.ppt"')
```

Commande *WINSERTBRK*

Insère un saut de page, une colonne ou une section dans un document Word.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WINSERTBRK')
CHGVAR          VAR(&PARM1) VALUE('[Break=Constante]')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Break = type de saut à insérer (Défaut= wdPageBreak).</p> <p>Les valeurs possibles sont les suivantes :</p> <p><i>wdPageBreak, wdColumnBreak, wdSectionBreakNextPage, wdSectionBreakContinuous, wdSectionBreakEvenPage, wdSectionBreakOddPage, wdLineBreak, wdLineBreakClearLeft, wdLineBreakClearRight, wdTextWrappingBreak.</i></p> |

Exemple

```
CHGVAR          VAR(&CMD) VALUE('WINSERTBRK')
CHGVAR          VAR(&PARM1) VALUE('Break=wdColumnBreak')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Commande *WINSERTCOL*

Insère une ou plusieurs colonnes à gauche dans un tableau Word.

Le curseur doit se trouver dans un tableau dans le document.

Les colonnes sont insérées à gauche de la colonne où se trouve le curseur.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WINSERTCOL')
CHGVAR          VAR (&PARM1) VALUE (' [NbCols] ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID (LNC0000) EXEC (GOTO CMDLBL (ERROR))
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Nombre de colonnes à insérer (par défaut 1 colonne). |

Exemple

```
CHGVAR          VAR (&CMD) VALUE ('WINSERTCOL')
CHGVAR          VAR (&PARM1) VALUE ('4')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID (LNC0000) EXEC (GOTO CMDLBL (ERROR))
```

Voir aussi

- [WINSERTROW](#)
- [WINSERTIMG](#)

Commande *WINSERTF*

Insère un document Word déjà existant dans le document Word actif.

La mise en forme du texte du document inséré reste identique.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WINSERTF')
CHGVAR          VAR(&PARM1) VALUE('
File="Nom du document"
[;Directory="Chemin de répertoire"]
[;Bookmark="Signet"]
[;Adjust=True/False]
[;Append=True/False]
[;range="plage"]
[;Link=True/False]
[;DeleteBookmark=True/False/"Nom"]
[;AsText=True/False]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>File : Nom du fichier à insérer. Si le chemin complet est spécifié, alors le paramètre Directory n'est pas utile.</p> <p>Directory : (Optionnel) Cette option permet de spécifier un chemin vers le répertoire où se trouve le document à insérer. Ce chemin de répertoire est ajouté devant le nom de fichier spécifié avec l'option File. Ce paramètre n'est pas nécessaire si le chemin complet est spécifié dans le paramètre File.</p> <p>Bookmark : (Optionnel) Nom du signet à l'emplacement duquel le nouveau document doit être inséré. On peut utiliser aussi un signet prédéfini de Microsoft Word.</p> <p>Adjust : (Optionnel) Si le positionnement à un signet est demandé par l'option Bookmark, cette option replace le signet sur l'ensemble du texte inséré.</p> <p>Append : (Optionnel) Si le positionnement à un signet est demandé par l'option Bookmark, cette option permet d'ajouter au contenu actuel du signet, le contenu du document à insérer.</p> |

Link = True / False, (False par défaut).

Si **Link** est vrai, c'est un lien vers le fichier qui est inséré. Les modifications apportées au fichier seront automatiquement reportées sur les documents avec lequel il est lié.

Range : Plage à insérer du document source. Si le fichier à insérer est un document Word, **Range** peut représenter un signet. Si le fichier est un document Excel, **Range** peut représenter une plage de cellules.

DeleteBookmark: Si cette option a la valeur Vrai, alors le signet spécifié par l'option **BookMark** est supprimé.

Si cette option contient une valeur alpha numérique, alors les signets seront supprimés comme décrit par la commande [WBOOKMDEL](#).

Si le document inséré contient lui même des signets, ces derniers seront ajoutés au document courant, si leur noms n'y existent pas déjà.

Si **AsText** est vrai, alors le fichier à insérer doit être de type texte seulement. Ce dernier sera alors inséré dans le document comme s'il était frappé au clavier. Le format de caractères et de paragraphes sera celui appliqué à l'endroit de l'insertion.

Exemples

Insertion d'un document Word à la fin du document Word actif :

```
CHGVAR      VAR(&CMD) VALUE('WINSERTF')
CHGVAR      VAR(&PARM1) VALUE('File="c:\temp\lettre.docx";+
                          Bookmark="\ENDOFDOC')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                          &PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Insertion d'un document Word à l'emplacement du signet "Etat" :

```
LNCCMD      CMD(WINSERTF) +
            PARM1('FILE="C:\temp\etat.docx";BOOKMARK="Etat"')
```

Commande *WINSERTIMG*

Insère une image dans le document Word courant.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WINSERTIMG')
CHGVAR      VAR(&PARM1) VALUE('
File="Nom du fichier image"
[;Directory="Chemin de répertoire"]
[;Bookmark="Signet"]
[;Adjust=True/False]
[;Append=True/False]
[;Link=True/False]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>File : Nom du fichier image à insérer. Si le chemin complet est spécifié, alors le paramètre Directory n'est pas utile.</p> <p>Directory : (Optionnel) Cette option permet de spécifier un chemin vers le répertoire où se trouve le fichier à insérer. Ce chemin de répertoire est ajouté devant le nom de fichier spécifié avec l'option File. Ce paramètre n'est pas nécessaire si le chemin complet est spécifié dans le paramètre File.</p> <p>Bookmark : (Optionnel) Nom du signet à l'emplacement duquel l'image doit être insérée. On peut utiliser aussi un signet prédéfini de Microsoft Word.</p> <p>Adjust : (Optionnel) Si le positionnement à un signet est demandé par l'option Bookmark, cette option replace le signet sur l'image insérée.</p> <p>Append : (Optionnel) Si le positionnement à un signet est demandé par l'option Bookmark, cette option permet d'ajouter au contenu actuel du signet, l'image à insérer.</p> <p>Link = True / False, False par défaut. Si Link est vrai, c'est un lien vers l'image qui est inséré. Les modifications apportées à l'image seront automatiquement</p> |

reportées sur les document avec lequel elle est liée.

Exemple

```
LNCCMD CMD (WORDOPEN)
```

```
LNCCMD CMD (WOPENFILE) +  
PARM1 ('%LNCDIR%\SAMPLES\MODELE_STYLE.DOCX')
```

```
LNCCMD CMD (WINSERTIMG) +  
PARM1 ('FILE="%LNCDIR%\SAMPLES\LNC_Ofc.jpg";+  
bookmark="image";link=true')
```

Autre exemple

```
LNCCMD CMD (WINSERTIMG) +  
PARM1 ('FILE="C:\A\GRAPHE_IMMO.JPG";BOO+  
KMARK="GRAPHE_IMMO"')
```

Voir aussi

- [WINSERTF](#)

Commande *WINSERTROW*

Insère une ou plusieurs lignes au dessus dans un tableau WORD.

Le curseur doit se trouver **dans un tableau** dans le document.

Les lignes sont insérées au dessus de la ligne où se trouve le curseur.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WINSERTROW')
CHGVAR          VAR(&PARM1) VALUE('
                [Count=Nombre lignes]
                [;MoveStart=True/False]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Count=Nombre de lignes à insérer. (<u>1 ligne par défaut</u>).</p> <p>MoveStart : Si True, la sélection courante sera déplacée au début des lignes insérées.</p> <p>Par défaut, la sélection courante reste sur la ligne en cours, c'est à dire, celle qui suit les lignes insérées.</p> |

Exemple

```

CHGVAR          VAR(&CMD) VALUE('WINSERTROW')
CHGVAR          VAR(&PARM1) VALUE('Count=20')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Voir aussi

- [WINSERTCOL](#)

Commande *WINSERTTBL*

Insère un tableau dans le document Word courant.

Le tableau peut être inséré à la position courante ou bien à l'emplacement d'un signet.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WINSERTTBL')
CHGVAR          VAR(&PARM1) VALUE('
Columns=Nombre de colonnes
;Rows=Nombre de lignes
[;Bookmark="Nom signet"]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Columns=nombre de colonnes du tableau.</p> <p>Rows=nombre de lignes du tableau.</p> <p>Bookmark=nom du signet où l'on souhaite insérer le tableau. Par défaut le tableau est inséré à la position courante.</p> |

Exemple

```

CHGVAR          VAR(&CMD) VALUE('WINSERTTBL')
CHGVAR          VAR(&PARM1) VALUE('Bookmark="table";Rows=5;Columns=6')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Voir aussi

- [WGOTOTBL](#)
- [WINSERTROW](#)
- [WINSERTCOL](#)

Commande WMAILMERGE

Exécute une fusion entre le modèle Word et le fichier de données.

Cette commande utilise les champs de fusion de Word.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WMAILMERGE')
CHGVAR      VAR(&PARM1) VALUE('
[Document="nom document principal"]
[;Destination=constante]
[;SaveAs="nom document sauvegarde"]
[;SaveAsFmt=constante]
[;DataSource="nom source données"]
[;HeaderSource="nom fichier entêtes"]
[;FirstRecord= Valeur]
[;LastRecord= Valeur]
[;SuppressBlankLines=True/False]
[;SourceFormat=Valeur]
[;HeaderFormat=Valeur]
[;Type=constante]
[;Execute=True/False ]
[;MailAddress="Champs adresse destinataire"]
[;MailSubject="sujet du message"]
[;AsAttachment=True/False]
[;OneDoc=True/False]
[;CleanData=True/False]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Document : nom du document principal, si ce paramètre est vide le document courant sera considéré comme document principal. S'il a été initialisé par une première opération de fusion, il est conservé.</p> <p>Destination : destination de la fusion, peut-être une des constantes suivantes :</p> <ul style="list-style-type: none"> wdSendToEmail, pour un envoi par mail. Renseignez alors le paramètre MailAddress. wdSendToFax, pour un envoi par fax. wdSendToNewDocument, pour créer un nouveau document (valeur par défaut). Renseignez alors le paramètre SaveAS. wdSendToPrinter, pour un envoi sur l'imprimante. |

IncSendToFile, pour sauver chaque lettre vers un fichier individuel. Le paramètre **SaveAs** doit alors contenir un chemin et des règles de construction d'un nom de fichier unique pour chaque lettre.

SaveAs : Chemin et nom du fichier où sauvegarder le résultat de la fusion.

Si **Destination=IncSendToFile**, ou **OneDoc=true**, alors le nom du fichier spécifié par le paramètre **SaveAs**, doit être construit à partir de valeurs de champs de la source de donnée.

SaveFmt : wdFormatXMLDocument / wdFormatDocument / wdFormatDOSText / wdFormatDOSTextLineBreaks / wdFormatEncodedText / wdFormatHTML / wdFormatRTF / wdFormatTemplate / wdFormatText / wdFormatTextLineBreaks / wdFormatUnicodeText.

Si ce paramètre n'est pas spécifié, c'est le format par défaut de Word qui est utilisé.

Par exemple, pour Word 2019, il s'agit du format wdFormatXMLDocument (DOCX).

Pour sauver au format **PDF** (à partir de Word 2010) on peut utiliser la valeur **17**.

DataSource : nom du fichier source. Ce paramètre n'est pas obligatoire si le document principal contient déjà le nom de la source de données associée.

HeaderSource : nom du fichier qui contient les entêtes. Permet de spécifier une source distincte qui contient le nom des entêtes de colonnes (et qui apparaissent dans le document principal comme champs de fusion).

FirstRecord : numéro du premier enregistrement de la source de données, à utiliser pour le publipostage.

Si ce paramètre n'est pas spécifié, le premier enregistrement de la source de données est pris en compte.

LastRecord : numéro du dernier enregistrement de la source de données, à utiliser pour le publipostage.

Si ce paramètre n'est pas spécifié, le dernier enregistrement de la source de données est pris en compte.

SuppressBlankLines, indique s'il faut conserver les lignes vides ou non, après la fusion. Par défaut : true.

SourceFormat, **HeaderFormat** indiquent respectivement le format du fichier source et du fichier d'entête.

Les valeurs possibles sont les suivantes :

wdOpenFormatAllWord, wdOpenFormatAuto, wdOpenFormatDocument, wdOpenFormatEncodedText, wdOpenFormatRTF, wdOpenFormatTemplate, wdOpenFormatText, wdOpenFormatUnicodeText, or wdOpenFormatWebPages.

La valeur par défaut est : *wdOpenFormatAuto*.

Type : type de document de fusion. Les valeurs possibles sont les suivantes :

wdCatalog, wdEnvelopes, wdFormLetters (par défaut), wdMailingLabels, wdNotAMergeDocument.

Execute (Par défaut *execute=true*) : Si "*execute = false*", le document est simplement préparé pour une fusion, mais pas fusionné.

Onedoc=true équivaut à "**Destination= IncSendToFile**", pour sauver chaque lettre vers un fichier individuel. Le paramètre **SaveAs** doit alors contenir un chemin et des règles de construction d'un nom de fichier unique pour chaque lettre. Par défaut : *false*.

CleanData : (Par défaut *False*) : Si *True*, on supprime les caractères non imprimables suivants du fichier de données : NUL, SI et CR (pas CRLF).

Si la destination est un envoi par email (**wdSendToEmail**), alors les propriétés suivantes peuvent être fixées :

MailAddress : Désigne le champ du fichier **DataSource** contenant l'adresse du destinataire.

MailSubject : Sujet du message email.

AsAttachment : indique que le document doit être attaché au message. Par défaut : *false*.

Remarques

1) **Aucun paramètre n'est obligatoire.**

2) Si le document principal a été préparé pour une fusion, il contient le nom de la source de donnée et la destination, il peut alors être ouvert par la commande

WDOCUMENT avant d'utiliser la commande WMAILMERGE sans aucun paramètres.

Le type de document principal permet de modifier les caractéristiques de la fusion, les différents types sont : lettres-types, étiquettes, enveloppes, catalogue (pas de sauts de page entre deux enregistrements). Il doit être défini lors de la préparation du document principal.

Si une première opération a initialisé les valeurs du document principal, de la source de données et de la destination, on peut relancer un WMAILMERGE qui reprendra tous ces paramètres. Par exemple pour fusionner uniquement un enregistrement ou une plage d'enregistrements particuliers.

3)

La propriété « Execute=False » permet de préparer par programme un document de fusion, en lui attachant la source de données, sans l'exécuter.

Une fois le document de fusion préparé, éventuellement vide, le programme peut le présenter à l'utilisateur (WORDSHOW) afin que ce dernier le compose.

4)

Construire un nom de fichier à partir d'un champ de la source de données :

Quand l'option **Destination=IncSendToFile** est spécifiée, ou bien quand **OneDoc=true**, chaque lettre est sauvée vers un fichier. Le chemin et le nom de fichier données par le mot clé **SaveAs** peut contenir des références à des champs de la source de donnée. Le champ auquel on fait référence doit être entre caractères < et >.

Exemple :

```
Destination=IncSendToFile;SaveAs="D:\Lettres\Prop_ <CUST_ID>.DOCX";
```

Dans cet exemple, chaque lettre sera sauvée dans le sous répertoire « Lettres », et le nom sera composé du mot « Prop_ » suivi de la valeur du champ « CUST_ID ».

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WMAILMERGE')
CHGVAR      VAR(&PARM1) VALUE('Document="C:\chemin\maitre.docx" ;+
                             Destination="wdSendToEmail";+
                             MailAddress="tech@easycom-aura.com"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                             &PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Fusionne le document principal "Maitre.docx" avec la source de donnée qui lui est déjà associée et envoie le résultat par mail au service technique d'AURA Equipements.

```

CHGVAR      VAR(&CMD) VALUE('WMAILMERGE')
CHGVAR      VAR(&PARM1) VALUE('Document="C:\chemin\maitre.docx" ;+
DataSource="C:\chemin\SPCUST.txt";+
SaveAs="result_5_6.docx";FirstRecord=5;+
LastRecord=6')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

CHGVAR      VAR(&CMD) VALUE('WMAILMERGE')
CHGVAR      VAR(&PARM1) VALUE('SaveAS="Result_17_20.docx"+
FirstRecord=17;LastRecord=20')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

```

Prépare la fusion de la source de donnée spcust.txt avec le document principal, fusionne les enregistrements 5 et 6 vers le document result_5_6.docx puis fusionne, avec les mêmes éléments, les enregistrements 17 à 20.

```

LNCCMD      CMD(WMAILMERGE) +
            PARM1('DOCUMENT="C:\temp\test\+
            sp_cust.dotx";DATASOURCE="C:\temp\test+
            \csv2.TXT";SAVEAS="C:\temp\res\+
            res_<CUST_ID>.pdf";OneDoc=true;SaveFmt=17')

```

Exécute la fusion en utilisant le modèle et la source de données spécifiés. Un PDF sera généré (basé sur le modèle) pour chaque enregistrement.

Voir aussi

- [LNCPRTDOC](#)
- [DBFXFER](#)
- [Créer un fichier de fusion](#)

Commande WMAXIMIZE

Permet de maximiser la fenêtre Word courante.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WMAXIMIZE')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WMINIMIZE](#)

Commande **WMENU**

Construit une barre de menu personnalisée sous Word.

WMENU doit être utilisé avec les commandes **WADDINS** et **WORDWAIT**.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WMENU')
CHGVAR      VAR(&PARM1) VALUE('
Captions="Libellés des boutons"
[;Tips="Bulles d'aide"]
[;Remove]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Captions : Permet de donner le libellé de chaque bouton de la nouvelle barre de menu. Les libellés sont séparés entre eux par un point virgule (;). Le nombre de libellés détermine le nombre de boutons de la barre de menu.</p> <p>Tips : Permet d'afficher une bulle d'aide pour chaque bouton. Les tips sont séparés entre eux par un point virgule (;). Leur nombre doit correspondre au nombre de libellés.</p> <p>Remove : supprime le précédent menu.</p> |

Remarques

1)

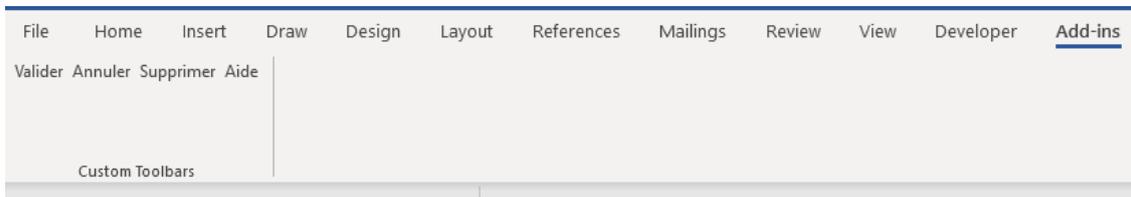
Le complément « LNCWordAddin.dot » doit être chargé par la commande **WADDINS** avant d'utiliser **WMENU**.

Lors de l'appel de la commande **WORDWAIT**, le programme AS/400 se met en attente d'une action de la part de l'utilisateur. Il reprend la main quand Word est fermé, ou sur l'action d'un des boutons du nouveau menu.

Au retour de la commande **WORDWAIT**, la variable &RESULT contient le numéro sur 5 chiffres du bouton actionné (de 1 à n).

2)

Le menu personnalisé apparaîtra dans l'onglet "Complément" du menu Word.



Exemple

```

CHGVAR          VAR (&CMD) VALUE ('WADDINS')
CHGVAR          VAR (&PARM1) VALUE ('%LNCDIR%\ LNCWordAddin.dot')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT)

CHGVAR          VAR (&CMD) VALUE ('WMENU')
CHGVAR          VAR (&PARM1) VALUE ('Captions="Valider;Annuler";Tips="Valider le
                        document;Abandonner le document"')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT)

CHGVAR          VAR (&CMD) VALUE ('WORDWAIT')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT)

```

Commande *WMERGECELL*

Fusionne plusieurs cellules d'un tableau Word.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WMERGECELL')
CHGVAR      VAR(&PARM1) VALUE('
             Cols=Nombre colonnes
             ;Rows=Nombre lignes
             ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
             &PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Cols : Nombre de colonne à sélectionner à partir de la position courante. Rows : Nombre de ligne à sélectionner à partir de la position courante. |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WMERGECELL')
CHGVAR      VAR(&PARM1) VALUE('cols=4;rows=2 ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
             &PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WSPLITCELL](#)

Commande WMETHOD

Appelle une méthode de l'objet Word.Application.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WMETHOD')
CHGVAR          VAR (&PARM1) VALUE ('Méthode')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID (LNC0000) EXEC (GOTO CMDLBL (ERROR))
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Le chemin complet de la méthode à appeler. Exemple : Documents.Item("Document2").Activate |
| Parm2 | |

Remarque 1

La commande **WMETHOD** permet d'exécuter depuis l'AS/400 des méthodes du langage VBA de Word.

Le contenu de Parm1 respecte la syntaxe utilisée en Visual Basic.

Il est possible d'utiliser toutes les constantes de Word et Visual Basic Application ou bien la valeur de la constante elle-même.

Pour avoir une idée des méthodes à appeler, placez vous sous Word en mode "Enregistrement de macros" : Menu **"Outils" – "Macro" – "Nouvelle macro"**.

Faites les opérations désirées au clavier et souris.

Arrêtez l'enregistrement de macro, et allez voir le code généré par Word : Menu **"Outils" – "Macro" – "Macros" – "Modifier"**.

Remarque 2

Il y a 2 différences à noter entre la syntaxe VB de Word et la syntaxe LAUNCHER :

1) Quand on sélectionne un objet d'une collection, il faut ajouter ".Item" après le nom de la collection.

```
Documents ("Document2").Activate
```

devient

```
Documents.Item ("Document2").Activate
```

2) Les noms de paramètres (Nom:=valeur) est propre à la syntaxe VB.

Avec LAUNCHER, il faut lister les valeurs de chaque paramètre, dans l'ordre attendu, séparés par ';'.

L'instruction suivante

```
ActiveDocument.Hyperlinks.Add Anchor:=Selection.Range, Address:=
-
    "http://www.easycom-aura.com", SubAddress:="",
ScreenTip:="", _
    TextToDisplay:="fffff"
```

Devient

```
ActiveDocument.Hyperlinks.Add(prop(Selection.Range) ;
http://www.easycom-aura.com; "" ; "" ; "Aura")
```

Et donc la commande est la suivante :

```
LNCCMD CMD(WMETHOD) +
PARM1 ('ActiveDocument.Hyperlinks.Add(prop(Selection.Range);+
"http://www.easycom-aura.com"; "" ; "" ; "Aura")')
```

Sinon, optez pour l'écriture d'une macro sous Word, que vous appellerez par [WEXEMACRO](#).

Exemple

L'exemple ci-dessous active le document "Document2".

```
CHGVAR          VAR (&CMD) VALUE ('WMETHOD')
CHGVAR          VAR (&PARM1) VALUE ('Documents.Item("Document2").Activate')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
```

Commande WMINIMIZE

Permet de minimiser la fenêtre Word courante.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WMINIMIZE')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WMAXIMIZE](#)

Commande WMOVE

Déplace ou étend la sélection courante.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WMOVE')
CHGVAR      VAR(&PARM1) VALUE('
[From="Signet"]
[;Direction="Sens du déplacement"]
[;Unit=Unité de déplacement]
[;Count=nombre d'unités du déplacement]
[;While="Liste de caractères"]
[;Until="Liste de caractères"]
[;Extend=True/False]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>From : Désigne un signet dans le document, où la sélection doit commencer. Si « From » est omis, le début de la sélection reste inchangé.</p> <p>Direction : Indique le sens du déplacement de la sélection courante. Le déplacement est opéré en fonction de la propriété Unit. Les valeurs possibles sont :</p> <ul style="list-style-type: none"> Start: Déplace la sélection au début de l'unité. End: Déplace la sélection à la fin. Down: Déplace la sélection vers le bas. Up: Déplace la sélection vers le haut. Right: Déplace la sélection vers la droite. Left: Déplace la sélection vers la gauche. <p>Voir la partie Détails pour utiliser correctement les autres paramètres ci-dessous en fonction du paramètre Direction.</p> <p>Unit : Indique l'unité de déplacement. Les valeurs possibles sont : wdCharacter, wdCell, wdWord, wdSentence, wdParagraph, wdSection, wdColumn, wdRow, wdTable, wdLine.</p> <p>Count : Indique le nombre d'unités du déplacement. Ce nombre peut être négatif ou positif.</p> <p>While : Tant que un des caractères de la liste est présent, la</p> |



sélection est déplacée.

Until : La sélection est déplacée jusqu'à ce qu'un des caractères de la liste soit rencontré.

Extend : Doit être à la valeur « True » pour que la sélection soit étendue, sinon, la sélection courante est remplacée.

Détails

| Paramètre Direction | Paramètres utilisables |
|---------------------|---|
| pas utilisé | <p>while Déplace la sélection spécifiée tant que l'un des caractères spécifiés est trouvé dans le document.</p> <p>[count] valeur par défaut: wdForward. Nombre maximal de caractères dont la sélection spécifiée doit être déplacée. Peut être un nombre ou la constante wdForward ou wdBackward. Si Count est un nombre positif, la sélection spécifiée est avancée dans le document, à partir de la fin. S'il s'agit d'un nombre négatif, la sélection est déplacée vers l'arrière, à partir de la position de départ.</p> <p>Exemple 1.</p> |
| | <p>until Déplace la sélection spécifiée jusqu'à ce que l'un des caractères spécifiés soit trouvé dans le document.</p> <p>[count] valeur par défaut: wdForward. Nombre maximal de caractères dont la sélection spécifiée doit être déplacée. Peut être un nombre ou la constante wdForward ou wdBackward. Si Count est un nombre positif, la sélection spécifiée est avancée dans le document, à partir de la fin. S'il s'agit d'un nombre négatif, la sélection est déplacée vers l'arrière, à partir de la position de départ.</p> <p>Exemple 2.</p> |
| | <p>Réduit la sélection spécifiée à sa position de début ou de fin, puis déplace l'objet réduit du nombre d'unités spécifié. Cette méthode renvoie une valeur Long qui représente le nombre d'unités avec lesquelles la sélection a été déplacée, ou renvoie 0 (zéro) si le déplacement a échoué.</p> |



| | |
|---------------------|---|
| | <p>[unit] valeur par défaut: wdCharacter Unité avec laquelle on va déplacer la position du caractère de fin.</p> <p>[count] valeur par défaut: 1. Nombre d'unités dont la plage ou la sélection spécifiée doit être déplacée. Si Count est un nombre positif, l'objet est réduit à sa position finale et déplacé vers l'arrière dans le document selon le nombre d'unités spécifié. Si Count est un nombre négatif, l'objet est réduit à sa position de départ et avancé du nombre d'unités spécifié.</p> <p>Exemple 3.</p> |
| <p>start</p> | <p>while Déplace la position de départ de la sélection spécifiée lorsque l'un des caractères spécifiés est trouvé dans le document.</p> <p>[count] Nombre maximal de caractères dont la plage spécifiée doit être déplacée. Peut être un nombre ou la constante wdForward ou wdBackward. Si Count est un nombre positif, la plage est avancée dans le document. S'il s'agit d'un nombre négatif, la plage est reculée. La valeur par défaut est wdForward.</p> <p>Exemple 4.</p> |
| | <p>until Déplace la position de départ de la sélection spécifiée jusqu'à ce que l'un des caractères spécifiés soit trouvé dans le document. Si le mouvement est en arrière dans le document, la sélection est développée.</p> <p>[count] Nombre maximal de caractères par lequel la plage spécifiée doit être déplacée. Peut être un nombre ou la constante wdForward ou wdBackward. Si Count est un nombre positif, la plage est avancée dans le document. S'il s'agit d'un nombre négatif, la plage est reculée. La valeur par défaut est wdForward.</p> <p>Exemple 5.</p> |
| | <p>Déplace la position de départ de la sélection spécifiée.</p> <p>[unit]</p> |

| | |
|-------------------|--|
| | <p>Unité par laquelle la position de départ de la sélection spécifiée doit être déplacée. Peut être l'une des constantes WdUnits. La valeur par défaut est wdCharacter.</p> <p>[count] Nombre maximal d'unités par lesquelles la plage spécifiée doit être déplacée. Si Count est un nombre positif, la position de départ de la plage est avancée dans le document. S'il s'agit d'un nombre négatif, la position de départ est reculée. Si la position de départ est avancée au-delà de la position de fin, la plage est réduite et les positions de début et de fin sont déplacées ensemble. La valeur par défaut est 1.</p> <p>Exemple 6</p> |
| <p>End</p> | <p>while Déplace la position du caractère de fin d'une plage lorsque l'un des caractères spécifiés est trouvé dans le document.</p> <p>[count] Nombre maximal de caractères par lequel la plage doit être déplacée. Peut être un nombre ou la constante wdForward ou wdBackward. Si Count est un nombre positif, la plage est avancée dans le document. S'il s'agit d'un nombre négatif, la plage est reculée. La valeur par défaut est wdForward.</p> <p>Exemple 7.</p> |
| | <p>until Déplace la position de fin de la sélection spécifiée jusqu'à ce que l'un des caractères spécifiés soit trouvé dans le document.</p> <p>[count] Nombre maximal de caractères par lequel la sélection spécifiée doit être déplacée. Peut être un nombre ou wdForward ou wdBackward. Si Count est un nombre positif, la sélection est avancée dans le document. S'il s'agit d'un nombre négatif, la sélection est déplacée vers l'arrière. La valeur par défaut est wdForward.</p> <p>Exemple 8.</p> |
| | <p>Déplace la position du caractère de fin d'une plage ou d'une sélection.</p> <p>[unit] Unité avec laquelle on va déplacer la position du caractère de fin. La valeur par défaut est wdCharacter.</p> |

| | |
|---|---|
| | <p>[count] Le nombre d'unités à déplacer. Si ce nombre est positif, la position du caractère de fin est avancée dans le document. Si ce nombre est négatif, la fin est reculée. Si la position finale dépasse la position de départ, la plage se réduit et les deux positions se déplacent ensemble. La valeur par défaut est 1.</p> <p>Exemple 9.</p> |
| <p>down Déplace la sélection vers le bas et renvoie le nombre d'unités dont elle a été déplacée.</p> | <p>[unit] L'unité dont la sélection doit être déplacée. La valeur par défaut est wdLine.</p> <p>[count] Le nombre d'unités dont la sélection doit être déplacée. La valeur par défaut est 1.</p> <p>[extend]. Peut être false ou true. Si false est utilisé, la sélection est réduite à la fin et déplacée vers le bas. Si true est utilisé, la sélection est étendue vers le bas. La valeur par défaut est false .</p> <p>Exemple 10.</p> |
| <p>up Déplace la sélection vers le haut et renvoie le nombre d'unités dont elle a été déplacée.</p> | <p>[unit] Peut être l'une des constantes suivantes: wdLine, wdParagraph, wdWindow ou wdScreen. La valeur par défaut est wdLine. Vous pouvez utiliser la constante wdWindow pour que l'argument Unit se déplace en haut ou en bas de la fenêtre active. Quelle que soit la valeur de Count (supérieure à 1 ou inférieure à 1), la constante wdWindow ne déplace qu'une unité. Utilisez la constante wdScreen pour déplacer plusieurs écrans.</p> <p>[count] Le nombre d'unités dont la sélection doit être déplacée. La valeur par défaut est 1.</p> <p>[extend] Spécifie si la sélection est déplacée ou étendue. Peut être false ou true. Si false est utilisé, la sélection est réduite à la fin et déplacée vers le haut. Si true est utilisé, la sélection est étendue. La valeur par défaut est false.</p> <p>Exemple 11.</p> |
| <p>right Déplace la sélection vers la droite et renvoie le nombre</p> | <p>[unit] Unité avec laquelle la sélection doit être déplacée. La valeur par défaut est wdCharacter.</p> |

| | |
|--|---|
| <p>d'unités dont elle a été déplacée.</p> | <p>[count] Le nombre d'unités dont la sélection doit être déplacée. La valeur par défaut est 1</p> <p>[extend] Peut être false ou true. Si false est utilisé, la sélection est réduite à la fin et déplacée à droite. Si true est utilisé, la sélection est étendue à droite. La valeur par défaut est false</p> <p>Exemple 12.</p> |
| <p>left Déplace la sélection vers la gauche et renvoie le nombre d'unités dont elle a été déplacée.</p> | <p>[unit] Unité avec laquelle la sélection doit être déplacée. La valeur par défaut est wdCharacter.</p> <p>[count] Le nombre d'unités dont la sélection doit être déplacée. La valeur par défaut est 1</p> <p>[extend] Peut être false ou true. Si false est utilisé, la sélection est réduite à la fin et déplacée vers la gauche. Si true est utilisé, la sélection est étendue à gauche. La valeur par défaut est false.</p> <p>Exemple 13.</p> |

Exemple 1:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('While="Z"') : result = 0

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('While="R"'): le curseur est déplacé d'un caractère. Result = 1

LNCCMD CMD(WMOVE) PARM1('While="R";count=2') : le curseur est déplacé d'un caractère. Result = 1

GOOD MORNING CEDRICK. AURA

c)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('While=" DOM";count=-5') : return ==-4

GOOD MORNING CEDRICK. AURA

Exemple 2:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Until="D";count=20') : result=8. le curseur est déplacé de 8 caractères

GOOD MORNING CEDRICK. AURA

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Until="D";count=2'): result=0

Aucun caractère «D» trouvé dans la plage spécifiée par count = 2. La plage n'est pas modifiée et la méthode renvoie 0 (zéro).

Exemple 3:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) : result=1 . Un caractère déplacé.

GOOD MORNING CEDRICK. AURA

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Unit=wdWord') : result= 1. Un mot déplacé.

GOOD MORNING CEDRICK. AURA

c)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('count=4')

GOOD MORNING CEDRICK. AURA

d)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('count=-3')

GOOD MORNING CEDRICK. AURA

Exemple 4:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Start";While="Z") : result = 0

LNCCMD CMD(WMOVE) PARM1('Direction="Start";While="R") : result = 0. La plage commence par le caractère M.

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Start";While="MOR") : result = 3. Le début de la plage a été déplacé de 3 caractères.

GOOD MORNING CEDRICK. AURA

Exemple 5:

a)

GOOD MORNING CEDRICK. AURA



LNCCMD CMD(WMOVE) PARM1('Direction="Start";Until="D";count=20') : result=11. Le curseur est déplacé de 11 caractères à partir du début de la sélection.

GOOD MORNING CEDRICK. AURA

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Start";Until="D";count=2'): result=0

c)

GOOD MORNING CEDRICK. AURA

Aucun caractère «D» trouvé dans la plage spécifiée par count = 2. La plage n'est pas modifiée et la méthode renvoie 0 (zéro).

LNCCMD CMD(WMOVE) PARM1('Direction="Start";Until="O";count=2')

GOOD MORNING CEDRICK. AURA

Exemple 6:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Start"') : result=1 . Un caractère déplacé.

GOOD MORNING CEDRICK. AURA

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Start";Unit=wdWord') : result= 1. Un mot déplacé.

GOOD MORNING CEDRICK. AURA

Exemple 7:

a)



GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="End";While="Z") : result = 0

GOOD MORNING CEDRICK. AURA

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="End";While="R") : result =1. Un caractère déplacé.

GOOD MORNING CEDRICK. AURA

c)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="End";While="MOR") : result = 1. La fin de la plage a été déplacée d'un caractère.

GOOD MORNING CEDRICK. AURA

Exemple 8:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="End";Until="D";count=20') : result=9. Le curseur a avancé de 9 caractères à partir de la fin de la sélection.

GOOD MORNING CEDRICK. AURA

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="End";Until="D";count=2') : result=0

GOOD MORNING CEDRICK. AURA

Aucun caractère «D» trouvé dans la plage spécifiée par count = 2. La plage n'est pas modifiée et la méthode renvoie 0 (zéro).

Exemple 9:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="End"') : result=1 . Un caractère déplacé.

GOOD MORNING CEDRICK. AURA

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="End";Unit=wdWord') : result= 1. Un mot déplacé.

GOOD MORNING CEDRICK. AURA

Exemple 10:

a)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Down"') : result=1 . Une ligne déplacée.

GOOD MORNING CEDRICK. AURA

|

b)

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Down";Extend=true') : result=1 . Une ligne déplacée.

GOOD MORNING CEDRICK. AURA



Exemple 11:

HELLO

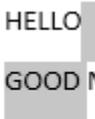
GOOD MORNING CEDRICK. AURA



LNCCMD CMD(WMOVE) PARM1('Direction="Up";Extend=true'): result=1 Une ligne déplacée.

HELLO

GOOD MORNING CEDRICK. AURA



Exemple 12:

HELLO

GOOD MORNING CEDRICK. AURA



LNCCMD CMD(WMOVE) PARM1('Direction="Right";Extend=true'): result=1 . Un caractère déplacé.

HELLO

GOOD MORNING CEDRICK. AURA



Exemple 13:

HELLO

GOOD MORNING CEDRICK. AURA

LNCCMD CMD(WMOVE) PARM1('Direction="Left";Extend=true'): result=1 . Un caractère déplacé.

HELLO

GOOD MORNING CEDRICK. AURA

Commande *WMOVERIGHT*

Déplace le curseur dans la cellule de droite dans un tableau Word.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WMOVERIGHT')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Remarque

Si le curseur est positionné dans la dernière cellule du tableau et qu'une instruction WMOVERIGHT est exécutée, une nouvelle ligne est ajoutée au tableau.

Commande *WNEWFILE*

Crée un document à partir d'un modèle Word.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WNEWFILE')
CHGVAR      VAR (&PARM1) VALUE ('modèle')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Chemin et Nom du modèle à utiliser pour créer un nouveau document. .dotx ou .docx doit être obligatoirement renseigné |
| Parm2 | |

Exemple

```
CHGVAR      VAR (&CMD) VALUE ('WNEWFILE')
CHGVAR      VAR (&PARM1) VALUE ('c:\temp\modele.dotx')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WOPENFILE](#)
- [WSAVE](#)
- [WSAVEAS](#)
- [WDOCUMENT](#)

Commande *WOPENFILE*

Ouvre un document **WORD**.

La commande **WORDOPEN** doit avoir été obligatoirement utilisée auparavant.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WOPENFILE')
CHGVAR      VAR(&PARM1) VALUE('fichier')
CHGVAR      VAR(&PARM2) VALUE('
[;*ONLY]
[*DOCPWD=xxx]
[*WRTPWD=yyy]
')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Chemin et nom du fichier à ouvrir. |
| Parm2 | Spécifiez *ONLY pour ouvrir le fichier en lecture seule. Spécifiez *DOCPWD=xxx pour donner le mot de passe nécessaire à l'ouverture du document. Spécifiez *WRTPWD=yyy pour donner le mot de passe nécessaire à la modification du document. |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WOPENFILE')
CHGVAR      VAR(&PARM1) VALUE('C:\temp\fichier.docx')
CHGVAR      VAR(&PARM2) VALUE('*ONLY')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WCLOSEFILE](#)
- [WSAVEAS](#)
- [WDOCUMENT](#)

Commande *WORDCLOSE*

Ferme Microsoft Word.

Les messages standards de sortie s'afficheront dans le cas où une intervention de l'utilisateur est nécessaire (impression en arrière plan, un fichier modifié non sauvegardé).

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WORDCLOSE')
CHGVAR          VAR(&PARM1) VALUE('save=true/false')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Save = True/False Si 'Save' n'est pas renseigné, Word va vous poser une question. « Voulez-vous enregistrer les modifications apportées à nom_document.docx ? » |

Voir aussi

- [WORDOPEN](#)

Commande *WORDHIDE*

Cache l'instance active de Word.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WORDHIDE')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG      MSGID (LNC0000) EXEC (GOTO CMDLBL (ERROR))
```

Voir aussi

- [WORDSHOW](#)
- [EXCELHIDE](#)

Commande WORDOPEN

Ouvre Microsoft WORD.

Un document peut être spécifié dans Parm1.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WORDOPEN')
CHGVAR      VAR(&PARM1) VALUE('NomDuFichier')
CHGVAR      VAR(&PARM2) VALUE('
[Directory="Chemin de répertoire";]
[Document="Nom de document";]
[Detach =True/False;]
[Visible;]
[ReadOnly;]
[New;]
[Minimize;]
[Maximize;]
[DOCPWD="Mot De Passe Ouverture";]
[WRTPWD="Mot De Passe Modifications";]
')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Nom du fichier à ouvrir, peut être vide pour la création d'un nouveau document ou contenir le nom d'un modèle si on spécifie NEW en paramètre 2 |
| Parm2 | <p>Directory Cette option permet de spécifier un chemin vers le répertoire où se trouve le document à ouvrir. Ce chemin de répertoire est ajouté devant le chemin et le nom de fichier donné en paramètre 1.</p> <p>Document Par le mot clé "Document", on demande à LAUNCHER Office de rechercher le document parmi ceux déjà ouverts par Word sur le système. Si le document demandé est déjà ouvert, il est rendu actif, il devient le document courant pour Word. Aucun nouveau fichier est ouvert, le nom du fichier ou du modèle sont alors ignorés.</p> <p>Detach = True/False (false par défaut) Si "Detach" est vrai, Word détache la source de données du document courant ou du nouveau document actif au retour de</p> |

l'appel de la commande WDOCUMENT. En détachant la source de données, le document n'est plus un modèle de publipostage pour Word. La commande WMAILMERGE refera l'attachement à une source de données.

Quand un document est ouvert et attaché à une source de données (Fichier de fusion), il est tenu par Word, et donc non modifiable. Il faut détacher la source de données du document courant si on veut envoyer des nouvelles données vers le fichier de fusion.

VISIBLE : pour rendre Word visible.

MINIMIZE : fenêtre réduite.

MAXIMIZE : fenêtre agrandie.

READONLY : ouvrir en lecture seule.

NEW : si Parm1 est vide, permet la création d'un nouveau document.

DOCPWD : mot de passe pour l'ouverture. Si le document est protégé par un mot de passe, il faut respecter la casse.

WRTPWD : mot de passe pour les modifications. Si le document est protégé contre les modifications, renseigner le mot de passe.

Attention !!! Word doit être visible pour ouvrir un document protégé par mot de passe.

RESULT

Nom du document

Au retour de la commande WDOCUMENT, le paramètre &RESULT contient le nom du document actif.

Pour un document ouvert depuis un fichier, le nom du document est égal au nom du fichier, sans le chemin et sans le suffixe ".DOCX". Si le document vient d'être créé, son nom a été construit par Word.

Exemple

```
LNCCMD CMD (WORDOPEN) PARM1 ('C:\A\Templates\sp_cust.docx')
```

Voir aussi

- [WORDCLOSE](#)
- [WDOCUMENT](#)

Commande *WORDSHOW*

Montre l'instance active de Word.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WORDSHOW')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 &PARM2+
&RESULT)
MONMSG         MSGID (LNC0000) EXEC (GOTO CMDLBL (ERROR))
```

Voir aussi

- [WORDHIDE](#)

Commande **WORDWAIT**

Attend un évènement Word.

Le programme AS/400 attend la fermeture de Word (**WORDCLOSE**), ou une action sur le menu personnalisé (commande **WMENU**), pour continuer son exécution.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WORDWAIT')
CHGVAR          VAR(&PARM1) VALUE('[BackPrint=Valeur] ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | La propriété BackPrint permet d'attendre qu'une impression en arrière plan se termine. La valeur est exprimée en secondes. 0 = infini. |

Voir aussi

- [WMENU](#)

Commande WPAGEBRK

Insère un saut de page à la position courante dans le document Word.

Dans un tableau, la ligne courante du point d'insertion est passée à la page suivante.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WPAGEBRK')
CHGVAR          VAR (&PARAM1) VALUE (' ')
CHGVAR          VAR (&PARAM2) VALUE (' ')

CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT))
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Commande WPASTE

Colle à la position courante du curseur, le contenu du presse-papiers.

La commande **WCOPY** peut être utilisée pour copier du texte dans le presse-papiers.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('WPASTE')
CHGVAR          VAR (&PARM2) VALUE (' ')
CHGVAR          VAR (&PARM1) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2+
&RESULT) )
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WCOPY](#)

Commande WPRINT

Permet d'imprimer le document WORD courant, en fixant un certains nombres de propriétés et d'options.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('WPRINT')
CHGVAR      VAR(&PARM1) VALUE('
                [Printer="Nom Imprimante"]
                [;TrayID=Numéro de bac]
                [;Tray="Nom du Bac"]
                [;Copies=Nombre De Copies]
                [;BackGround=True/False]
                [;Pages="Pages à Imprimer"]
                [;PageType=Type de pages]
                [;FirstPageTray= Numéro ou nom de bac]
                [;OtherPagesTray Numéro ou nom de bac]
                [;OutFile="Fichier de sortie"]
                [Execute=True/False]
                ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|----------------------|--|
| Parm1 ou Parm2 | <p>Printer est le nom de l'imprimante. Si ce paramètre est absent, l'impression se fait sur l'imprimante par défaut de Windows (voir Panneau de configuration).</p> <p>TrayID est un identifiant de bac. Les valeurs possibles sont listées plus bas.</p> <p>Tray est le nom du bac. Vous devez indiquer une chaîne telle qu'elle apparaît dans la liste des bacs des propriétés de l'imprimante.</p> <p>Copies est le nombre de copies à imprimer.</p> <p>BackGround : True/False. False par Défaut. Pour imprimer en arrière plan.</p> <p>PageType est le type de pages à imprimer, peut prendre une des valeurs constantes suivantes :</p> <p style="padding-left: 40px;"><i>wdPrintAllPages</i> : imprimer pages paires et impaires</p> <p style="padding-left: 40px;"><i>wdPrintEvenPagesOnly</i> : imprimer pages paires</p> |

wdPrintOddPagesOnly. : imprimer pages impaires

Pages est le nombre de pages et/ou l'intervalle de pages à imprimer, séparés par des virgules (,).
Par exemple, "2, 6-10" signifie : imprimer la page 2 et les pages 6 à 10.

FirstPageTray est l'identifiant ou le nom du bac pour la première page. Les valeurs possibles pour l'identifiant sont listées plus bas.

OtherPagesTray est l'identifiant ou le nom du bac pour les pages suivantes. Les valeurs possibles pour l'identifiant sont listées plus bas.

OutFile : Chemin complet vers un fichier de sortie qui recevra le flux d'impression.

Execute : Si "Execute" est faux, alors les propriétés du document sont changées (Copies, Pages, Bacs, ...) mais le document n'est pas imprimé.

Remarques

1)

Les paramètres sont tous optionnels.

2)

Valeurs possibles pour un identifiant de bac :

*wdPrinterAutomaticSheetFeed ; wdPrinterDefaultBin ;
wdPrinterEnvelopeFeedwd ; PrinterFormSource ; wdPrinterLargeCapacityBin ;
wdPrinterLargeFormatBin ; wdPrinterLowerBin ; wdPrinterManualEnvelopeFeed ;
wdPrinterManualFeed ; wdPrinterMiddleBin ; wdPrinterOnlyBin ;
wdPrinterPaperCassette ; wdPrinterSmallFormalBin ; wdPrinterTractorFeed ;
wdPrinterUpperBin*

Attention, ces constantes ne semblent pas être admises avec certaines imprimantes. Il vaut donc mieux utiliser le nom du bac.

Exemple

```
CHGVAR          VAR(&CMD) VALUE('WPRINT')
CHGVAR          VAR(&PARM1) VALUE('Printer="Brother HL-1270N +
series on Ne00"; Copies=2;+
TrayID=wdPrinterLowerBin;BackGround=False;+
Pages="2,4"')
```



```
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [IFSPUT](#)
- [CRTPRNSPLF - Commande CL](#)
- [SETPRINTER](#)

Commande WPROTECT

Protège le document Word courant contre les modifications.

Si un document est protégé, l'utilisateur ne peut lui apporter que des modifications limitées, comme ajouter des commentaires, effectuer des changements (ajouts et suppressions) signalés par des marques de suivi des modifications, ou remplir un formulaire.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WPROTECT')
CHGVAR          VAR(&PARM1) VALUE('
Level=Type de protection
[;Sections="Sections à protéger"]
[;PWD="Mot de passe"]
[;Exclude]
[;Add]
[;Remove]
[;UnProtect]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|----------------------|---|
| Parm1 ou Parm2 | <p>Level indique le niveau de protection à appliquer au document. Les valeurs possibles sont :</p> <ul style="list-style-type: none"> • wdAllowOnlyComments : Seul des commentaires peuvent être ajoutés. • wdAllowOnlyFormFields : Le document passe en mode saisie de formulaire. • wdAllowOnlyRevisions : Les modifications avec suivi sont autorisées. • wdNoProtection : Aucune protection. <p>PWD permet d'associer un mot de passe à la protection.</p> <p>UnProtect permet d'enlever la protection au document. Seule l'option PWD est utile pour enlever la protection, les autres options de la commande sont ignorées</p> <p>Les options suivantes ne s'appliquent que si Level est égal à wdAllowOnlyFormFields :</p> <ul style="list-style-type: none"> - Sections permet de choisir les sections à protéger, ou à exclure |



de la protection. Les numéros des sections sont données entre guillemets, séparés par virgule.

La valeur spéciale "***ALL**" indique que l'on veut protéger ou déprotéger toutes les sections.

La valeur spéciale "***CURRENT**" désigne la section dans laquelle se trouve la sélection courante.

Si le mot clé **Sections** est absent, la protection s'applique aux sections, tel qu'elle était avant une éventuelle suppression de protection.

- **Exclude** indique que les sections qui sont listées par le mot clé **Sections** ne doivent pas être protégées. Les autres sections seront protégées.
- **Add** permet de protéger les sections listées. La protection des autres sections reste inchangée.
- **Remove** enlève la protection des sections listées. La protection des autres sections reste inchangée.

Si les options **Exclude**, **Add** et **Remove** sont absentes, la protection est appliquée aux sections listées, et enlevée aux autres sections.

Remarque

Si le niveau **wdAllowOnlyFormFields** est appliqué à un document qui n'a pas de champs de formulaire, alors aucune modification ne sera possible sur le document.

Exemples

```
CHGVAR      VAR(&CMD) VALUE('WPROTECT')
CHGVAR      VAR(&PARM1) VALUE('Level=wdallowonlycomments')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

```
CHGVAR      VAR(&CMD) VALUE('WPROTECT')
CHGVAR      VAR(&PARM1) VALUE('Level=wdAllowOnlyFormFields')
CHGVAR      VAR(&PARM2) VALUE('Sections ="2,3,5"')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

```
CHGVAR      VAR(&CMD) VALUE('WPROTECT')
CHGVAR      VAR(&PARM1) VALUE('Level= wdAllowOnlyFormFields')
```



```
CHGVAR      VAR(&PARM2) VALUE('Sections="*CURRENT";Exclude')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WDPROTECT](#)

Commande WREFVAR

Ecrit la valeur d'une variable de document à l'emplacement du curseur, dans le document Word courant.

Cette variable peut avoir été créée ou modifiée en utilisant la commande **WSETVAR**.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WREFVAR')
CHGVAR          VAR(&PARM1) VALUE('
Text="Nom de variable"
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Text : Nom de la variable du document. La valeur de la variable sera écrite dans le document Word courant, à l'emplacement du curseur. |

Exemple

Dans l'exemple qui suit, on se place sur le signet "VAR7RES", et on lui affecte la valeur de la variable "VAR7" :

```
LNCCMD          CMD(WOPENFILE) PARM1('C:\A\Test.docx')
LNCCMD          CMD(WBOOKMARK) PARM1('BOOKMARK="VAR7RES"')
LNCCMD          CMD(WREFVAR) PARM1('Text="VAR7"')
```

Voir aussi

- [WSETVAR](#)

Commande *WREPLACE*

Recherche un texte dans le document Word actif et le remplace par une nouvelle valeur. Plusieurs propriétés et options peuvent être précisées.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WREPLACE')
CHGVAR          VAR(&PARM1) VALUE('
Find="Texte à rechercher"
[;ReplaceWith="Nouveau texte"]
[;MatchCase=True/False]
[;MatchWholeWord=True/False]
[;MatchWildcards=True/False]
[;MatchSoundsLike=True/False]
[;Replace=constante]
[;Forward=True/False]
[;Wrap=constante]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Find : le texte qui sera recherché.</p> <p>ReplaceWith : le texte de remplacement. Pour supprimer toutes les occurrences d'un mot ou d'un texte, indiquez pour cet argument la chaîne vide (""). Si aucune valeur n'est donnée, le texte trouvé est simplement sélectionné.</p> <p>MatchCase: True/False. Vrai si la recherche doit respecter la casse. Faux par défaut.</p> <p>MatchWholeWord : True/False. Vrai pour rechercher les occurrences constituant des mots entiers et non des parties de mots. Faux par défaut.</p> <p>MatchWildcards : True/False. Vrai pour utiliser les caractères génériques et opérateurs spéciaux. Par exemple : ^p pour une marque de paragraphe, ^t pour une tabulation... Faux par défaut.</p> |

MatchSoundsLike : True/False. Recherche les mots qui ressemblent phonétiquement mais avec une orthographe différente. Faux par défaut.

Replace : indique le nombre de remplacements à effectuer : un, tous ou aucun. Utilise les constantes suivantes : **wdReplaceAll** (par défaut), **wdReplaceNone**, ou **wdReplaceOne**.

Forward : True/False. Recherche en avant, vers la fin du document. True par Défaut.

Wrap :

WdFindAsk : Après remplacement dans la sélection, Microsoft affiche un message demandant si vous souhaitez rechercher dans le reste du document.

WdFindContinue (par défaut) : L'opération de remplacement continue après la sélection.

WdFindStop : l'opération de remplacement de termine dès que la fin de la sélection est atteinte.

Exemple

```
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('FILE="C:\TEMP\TEST.DOC"')
CHGVAR VAR(&PARM2) VALUE('VISIBLE')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR VAR(&CMD) VALUE('WREPLACE')
CHGVAR VAR(&PARM1) VALUE('FIND="Bonjour";+
REPLACEWITH="Hello"')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [WFINDTEXT](#)

Commande WSAVE

Sauvegarde les modifications apportées au document Word courant.

Si ce document n'a pas été sauvegardé auparavant, une boîte de dialogue apparaîtra pour demander le nom de sauvegarde.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('WSAVE')
CHGVAR          VAR(&PARM1) VALUE(' ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande WSAVEAS

Sauvegarde du document WORD courant.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WSAVEAS')
CHGVAR          VAR(&PARM1) VALUE('
File="nom du fichier"
[;RecentFiles=True/False]
[;Format=constante]
[;OptimizeSize=True/False]
[;PdfA=True/False]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>File : Chemin et nom de sauvegarde du fichier courant.</p> <p>RecentFiles : True/False. Si True, le fichier sauvegardé est ajouté à la liste des derniers fichiers utilisés. False par défaut.</p> <p>Format : Le format doit être cohérent avec l'extension du fichier sauvegardé. Vous pouvez utiliser la syntaxe Word : <i>wdFormatDocument, wdFormatDOSText, wdFormatDOSTextLineBreaks, wdFormatEncodedText, wdFormatHTML, wdFormatRTF, wdFormatTemplate, wdFormatText, wdFormatTextLineBreaks, wdFormatUnicodeText...</i> ou bien les mots clés suivants :</p> <ul style="list-style-type: none"> - DOC - DOCX - DOCM - PDF - HTML - XPS <p>Si vous ne mettez pas le paramètre Format, alors le format par défaut de MS Word est utilisé. Exemple : DOCX pour Word 2010 et versions ultérieures.</p> |

OptimizeSize : True/False. False par défaut.

Ce paramètre est pris en compte seulement si Format=PDF.

Si True, la résolution et la qualité du PDF sont optimisées pour un affichage à l'écran. La qualité est moindre et la taille est minimale.

Si False, le PDF est optimisé pour l'impression. La qualité est optimale et la taille du fichier est plus grande.

PdfA : True/False. False par défaut.

Ce paramètre est pris en compte seulement si Format=PDF.

Si True, le PDF sauvegardé sera compatible avec la norme PDF/A-3A (Nom ISO : ISO 19005-3).

Exemples

1)

```
LNCCMD CMD (WSAVEAS)
PARM1 ('File="c:\temp\test_wsaveas.docx";format=DOCX')
```

2)

```
LNCCMD CMD (WSAVEAS)
PARM1 ('File="c:\temp\test_wsaveas.docm";format=DOCM')
```

3)

```
LNCCMD CMD (WSAVEAS)
PARM1 ('FILE="C:\A\test.PDF";FORMAT=PDF;OptimizeSize=true')
```

Pour avoir une taille du PDF minimale, l'option **OptimizeSize** est égale à true.

4)

```
LNCCMD CMD (WSAVEAS)
PARM1 ('File="C:\temp\test\4\Doc3.pdf";Format=PDF;PdfA=true')
```

Pour avoir un PDF compatible avec la norme PDF/A-3A.

Voir aussi

- [WCLOSEFILE](#)
- [WOPENFILE](#)

Commande WSELECT

Sélectionne du texte dans un document Word.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WSELECT')
CHGVAR      VAR(&PARM1) VALUE('*WORD ou *SENTENCE ou *PARAGRAPH ou *LINE
ou *COLUMN ou *ROW ou *CELL ou *TABLE')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Type de sélection à effectuer : <ul style="list-style-type: none">*WORD : un mot*SENTENCE : une phrase*PARAGRAPH : un paragraphe*LINE : une ligne*COLUMN : une colonne (dans un tableau)*ROW : une rangée (dans un tableau)*CELL : une cellule (dans un tableau)*TABLE : un tableau complet |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WSELECT')
CHGVAR      VAR(&PARM1) VALUE('*TABLE')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [WCOPY](#)
- [WGETTEXT](#)

Commande *WSELROW*

Cette commande permet de sélectionner une ligne dans un tableau.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('WSELROW')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Commande WSETLINE

Affecte des valeurs à des cellules d'un tableau Word.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WSETLINE')
CHGVAR          VAR(&PARM1) VALUE('Valeurs de cellules')
CHGVAR          VAR(&PARM2) VALUE('Formats de cellules')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG          MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Chaîne de caractères contenant les valeurs à affecter aux cellules. Les valeurs de chaque cellule sont séparées par le mot %SEP% . La chaîne de caractères peut contenir les mots symboliques suivants :</p> <p>%SEP% : Pour passer à la cellule suivante. Si on était positionné sur la dernière colonne du tableau, on passe au début de la ligne suivante.</p> <p>%INS% : Pour insérer une nouvelle ligne, et se positionner en début de la nouvelle ligne.</p> <p>%MRG% : Pour fusionner la cellule courante avec la suivante.</p> |
| Parm2 | <p>Chaîne de caractères pouvant contenir les formats des cellules. Les formats de chaque cellule sont séparés par le mot %SEP%. On peut appliquer plusieurs formats sur une cellule. Ils sont alors séparés par un point virgule.</p> <p><u>Les formats peuvent être les suivants :</u></p> <p>NUMFMT(d [DECPOINT=p] [GRPPOINT=g])</p> <p>Où :</p> <ul style="list-style-type: none"> -d est le nombre de décimales, -p représente le caractère à utiliser pour le point décimal. (Caractère point par défaut), -g est le caractère à utiliser pour séparer les groupes de 3 chiffres. <p>PROP fixe la valeur d'une propriété pour le contenu de la cellule correspondante.</p> <p>Exemple :</p> <p><i>PROP(Font.bold)=True; PROP(Font.italic)=True</i></p> |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WSETLINE')
CHGVAR      VAR(&PARM1) VALUE('%INS%Valeur colonne 1%SEP%Valeur Colonne 2
& 3%MRG%%SEP%12300%SEP%')
CHGVAR      VAR(&PARM2) VALUE('%SEP%PROP(Font.Italic)=True;PROP(Par+
agraphs.Item(1).Alignment)=wdAlignParagraph+
Right%SEP%NUMFMT(2)')
```

Dans cet exemple, PARM1 contient les valeurs.

%INS% insère une nouvelle ligne,

« Valeur Colonne 1 » est affecté à la 1ère cellule.

« Valeur Colonne 2 & 3 » est affecté à la 2ème cellule.

Les cellules 2 et 3 sont fusionnées par %MRG%.

La valeur « 123,00 » est affecté à la 4^{ème} cellule.

Le dernier %SEP% déplace la position courante vers la cellule suivante.

Si le tableau ne contient que 4 colonnes, la position courante est déplacée au début de la ligne suivante.

PARM2 contient les formats :

PARM2 commence par %SEP%, donc la 1^{ère} cellule n'a pas de format particulier.

La deuxième cellule (Fusionnée à la 3^{ème}) sera en italique, et, cadrée à droite.

La 4^{ème} cellule est un numérique à deux décimales.

Voir aussi

- [Les valeurs spéciales](#)

Commande WSETVAR

Affecte une valeur à une variable du document Word courant.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('WSETVAR')
CHGVAR          VAR(&PARM1) VALUE('
                [Variable="Nom de variable"]
                [;Value="Valeur"]
                [;Update=True/False]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Variable : Nom de la variable du document. Si aucun nom n'est précisé, seule la propriété Update sera prise en considération.</p> <p>Value : Texte à affecter à la variable.</p> <p>Update : Permet de mettre à jour tous les champs et variables du document si True est spécifié.</p> |
| Parm2 | <p>Parm2 peut contenir la valeur à affecter à la variable si elle n'a pas été donnée dans Parm1.</p> |

Remarque

La variable de document Word, modifiée par la commande WSETVAR, peut ensuite être utilisée dans du code VBA.

Supposons que l'on exécute les commandes suivantes :

```
LNCCMD          CMD(WSETVAR) PARM1('variable="A";value="100"')
```

```
LNCCMD          CMD(WSETVAR) PARM1('VARIABLE="B";VALUE="3"')
```

En VBA, les variables peuvent être récupérées ainsi :

```
ActiveDocument.Variables.Item("A")
```

```
ActiveDocument.Variables.Item("B")
```

Exemple

Cet exemple affecte la valeur "Monsieur" à la variable "Civil".

```
CHGVAR          VAR(&CMD) VALUE('WSETVAR')
CHGVAR          VAR(&PARM1) VALUE('Variable="Civil";Value="Monsieur"')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [WREFVAR](#)

Commande WSETPASSWD

Permet de définir un mot de passe pour le document Word courant.

Il existe deux sortes de mots de passe :

- Un pour autoriser l'ouverture du document
- Un autre pour autoriser la modification du document

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WSETPASSWD')
CHGVAR      VAR(&PARM1) VALUE('
Type="Type de mot de passe"
;PWD="Mot de passe"
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Type désigne le type de mot de passe à ajouter au document: <ul style="list-style-type: none">• *DOCPWD pour spécifier le mot de passe à l'ouverture• *WRTPWD pour celui autorisant l'écriture. Le mot de passe est donné par le mot clé PWD . |

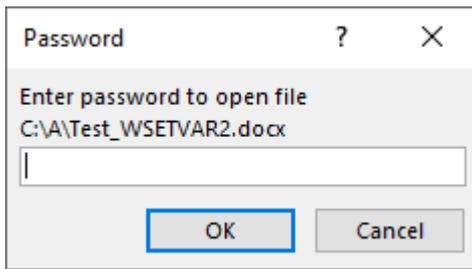
Exemple

```
LNCCMD      CMD(WOPENFILE) PARM1('C:\A\Test.docx')
```

```
LNCCMD      CMD(WSETPASSWD) PARM1('Type="*DOCPWD";PWD="aura"')
```

Remarque

L'ouverture d'un document Word protégé par un mot de passe, avec **WOPENFILE** par exemple, rendra Word visible et l'apparition d'une popup qui pourra gêner avec Launcher en mode service, si on ne spécifie par le mot de passe dans la commande.



Pour ne pas avoir de popup, il faut spécifier le mot de passe avec **WOPENFILE** :

```
LNCCMD      CMD(WOPENFILE) PARM1('C:\A\Test_WSETVAR2.docx') +  
            PARM2('*DOCPWD=aura')
```

Commande WSETPROP

Fixe la valeur d'une propriété sur le document Word courant ou sur l'application.

Syntaxe 1

```

CHGVAR          VAR (&CMD) VALUE ('WSETPROP')
CHGVAR          VAR (&PARM1) VALUE ('Propriété')
CHGVAR          VAR (&PARM2) VALUE ('Valeur')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT))
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Le chemin complet vers la propriété à modifier. Exemple : Selection.Font.Bold |
| Parm2 | Valeur à affecter à la propriété. <u>Exemples :</u> TRUE : Valeur Vrai. FALSE : Valeur Faux. INT(chaine) : Quand Word attend une valeur entière. |

Exemples 1

1) L'exemple ci-dessous fixe la couleur de police de la sélection en rouge.

```

CHGVAR          VAR (&CMD) VALUE ('WSETPROP')
CHGVAR          VAR (&PARM1) VALUE ('Selection.Font.Color')
CHGVAR          VAR (&PARM2) VALUE ('INT(255)')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                        &PARM2 &RESULT))
    
```

Remarque : Parm2 = INT(255) : Couleur rouge

2) Cet exemple insère du texte dans l'entête et pied de page.

```
LNCCMD CMD (WORDOPEN)
```

```
LNCCMD CMD (WOPENFILE) +  
PARM1 ('%LNCDIR%\SAMPLES\MODELE_STYLE.DOC')
```

```
LNCCMD CMD (WSETPROP) +  
PARM1 (activewindow.activepane.view.seekview+  
) PARM2 (wdseekcurrentpageheader)
```

```
LNCCMD CMD (WTYPETEXT) PARM1 ('CECI EST UNE ENTETE DE PAGE')
```

```
LNCCMD CMD (WSETPROP) +  
PARM1 (ACTIVIEWINDOW.ACTIVEPANE.VIEW.SEEKVIEW+  
) PARM2 (WDSEEKCURRENTPAGEFOOTER)
```

```
LNCCMD CMD (WTYPETEXT) PARM1 ('CECI EST UN PIED DE PAGE')
```

```
LNCCMD CMD (WORDSHOW)
```

3) Cet exemple met la sélection en gras.

```
LNCCMD CMD (WSETPROP) PARM1 ('Selection.Font.Bold') +  
PARM2 ('True')
```

4) Cet exemple désactive l'affichage de popups Word.

```
LNCCMD CMD (WSETPROP) +  
PARM1 ('application.displayalerts') +  
PARM2 ('wdalertsnone')
```

Syntaxe 2

```
CHGVAR VAR (&CMD) VALUE ('WSETPROP')  
CHGVAR VAR (&PARM1) VALUE ('  
Property= "Propriété";  
Value="Valeur"  
)  
CALL PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +  
&PARM2 &RESULT))
```



```
MONMSG          MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Property = Le chemin complet vers la propriété à modifier. |
| ou Parm2 | Value = Valeur à affecter à la propriété |

Exemple 2

```
LNCCMD          CMD(WSETPROP) PARM1('Property= "SELECTION.STYLE" ;Value="ACCENTUATION"') +
```

Remarque

Le **chemin vers la propriété** respecte la syntaxe utilisée en Visual Basic.

Quelques exemples :

Selection.Font.Color : Change la couleur de la sélection.

Selection.Borders.Item(wdBorderLeft).LineStyle: Change le style de la bordure gauche.

Selection.Font.Bold : Permet de mettre en gras la sélection.

Il est possible d'utiliser **toutes les constantes de Word et Visual Basic Application** ou bien la valeur de la constante elle-même.

Quand le chemin vers la propriété inclut un élément d'une collection (tel que Selection.Borders, qui a 4 bordures), il faut préciser l'élément concerné (Item).

Commande *WSPLITCELL*

Scinde la cellule courante d'un tableau Word en plusieurs cellules.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WSPLITCELL')
CHGVAR      VAR(&PARM1) VALUE('Nombre Cellules')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Nombre indiquant en combien de cellule va être séparée la cellule active. |
| Parm2 | |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('WSPLITCELL')
CHGVAR      VAR(&PARM1) VALUE('4')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [WMERGECELL](#)

Commande *WTYPETEXT*

Insère du texte dans un document Word, à l'emplacement du curseur.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('WTYPETEXT')
CHGVAR      VAR(&PARM1) VALUE('Texte')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Texte à insérer à l'emplacement du curseur. |
| Parm2 | |

Exemple

```
LNCCMD      CMD(WTYPETEXT) PARM1('Texte ajouté avec WTYPETEXT')
```

Commande **XLADDINS**

Ajoute un complément EXCEL à la liste des compléments (File, Options, Add-ins, Excel add-ins, Go). Le complément est également activé.

Un complément est un programme supplémentaire qui ajoute des commandes ou des fonctionnalités personnalisées à Microsoft EXCEL.

Un complément Excel est au format **.xla** (mode de compatibilité 2003) ou **.xlam**.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLADDINS')
CHGVAR      VAR(&PARAM1) VALUE('Complément')
CHGVAR      VAR(&PARAM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin complet vers le complément à installer. |
| Parm2 | |

Remarque

On peut utiliser **XLADDINS** pour ajouter un modèle qui contient des macros et les rendre ainsi disponibles.

Exemple

Chargement du complément fourni avec Launcher pour créer un menu personnalisé avec Launcher (commande **XLMENU**) :

```
CHGVAR      VAR(&CMD) VALUE('XLADDINS')
CHGVAR      VAR(&PARAM1) VALUE('%LNCDIR%\LNCEXCELADDIN.XLA')
CHGVAR      VAR(&PARAM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Voir aussi



- [XLMENU](#)
- [XLENAADD](#)
- [XLDISADD](#)

Commande XLADDSHEET

Ajoute une feuille au classeur courant.

La nouvelle feuille est ajoutée avant la feuille courante.

La nouvelle feuille devient la feuille courante.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('XLADDSHEET')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
```

Voir aussi

- [XLDELSHEET](#)
- [XLGOTOSH](#)
- [XLPRINTSH](#)
- [XLSHNAME](#)

Commande *XLBOLD*

Cette commande permet de mettre la valeur de la cellule en gras.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLBOLD')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLITALIC](#)
- [XLUNDERLN](#)

Commande XLCALCULAT

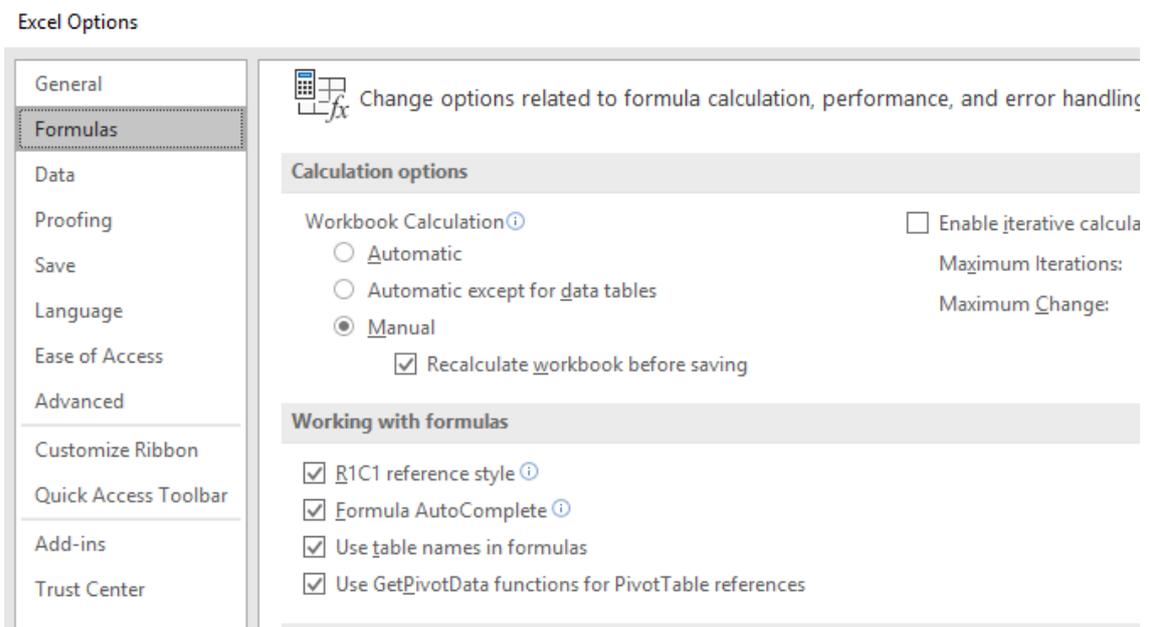
Calcule la feuille courante.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('XLCALCULAT')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Remarque

Cette commande ne marchera que si l'option "manuelle" est configurée pour les options de calcul d'Excel (F9 pour l'exécution manuelle).



Commande *XLCELLNAME*

Change le nom d'une cellule de la feuille courante.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('XLCELLNAME')
CHGVAR      VAR (&PARM1) VALUE ('coordonnées')
CHGVAR      VAR (&PARM2) VALUE ('nom')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Les coordonnées de la cellule à renommer . \$Colonne\$Ligne |
| Parm2 | Nouveau nom de la cellule. |

Exemple

```
CHGVAR      VAR (&CMD) VALUE ('XLCELLNAME')
CHGVAR      VAR (&PARM1) VALUE ('$A$3')
CHGVAR      VAR (&PARM2) VALUE ('Total')

CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLGOTOCELL](#)
- [XLSETTEXT](#)
- [XLSETVALUE](#)

Commande XLCELLS

Permet d'effectuer des actions sur une cellule ou un groupe de cellules.

- Lire la valeur
- Modifier la valeur
- Modifier la formule de la cellule
- Modifier le format
- Affecter un nom
- Modifier des propriétés
- Lire une propriété

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLCELLS')
CHGVAR          VAR(&PARM1) VALUE('
[Ref="Référence de cellules"]
[;Sheet="Nom de la feuille"]
[;Row= Numéro de ligne ou déplacement ]
[;Col= Numéro de colonne ou déplacement ]
[;RowCnt= Nombre de lignes sélectionnées ]
[;ColCnt= Nombre de colonnes sélectionnées ]
[;RefTo= "Référence de fin de zone sélectionnée" ]
[;NextOutLine="Niveaux"]
[;SetValue=True/False ]
[;SetFormula=True/False ]
[;Format="Format à affecter"]
[;Name="Nom à affecter"]
[;Comment="Commentaire"]
[;GetValue=True/False]
[;GetText=True/False]
[;GetProp("Propriété")]
[;GetSize=True/False]
[;Select=True/False]
[;Clear=True/False]
[;Freeze=True/False]
[;Calculate=True/False]
[;Merge=xlColumn|xlNone|xlAll]
[;Prop(propriétés)=Valeur de propriété]
[;DateFmt (Format date)]
[;NumFmt (Format nombres)]
[;IgnoreError=True/False]
[;Unicode=True/False]
')
CHGVAR          VAR(&PARM2) VALUE('Valeur affectée aux cellules')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

Paramètres



Parm1

Ref : La référence à la cellule peut être donnée sous la forme :

- Syntaxe **\$B\$3**. Exemple : \$B\$3 désigne la colonne B, ligne 3.
- Syntaxe **CL. C** pour désigner la colonne et **L** pour désigner la ligne.
Exemple : B3 désigne la ligne 3, et la colonne 2 (B).
- Nom symbolique donné à une cellule ou un groupe de cellules.
- "." désignera la cellule active.

Sheet : Indique le nom de la feuille à adresser.

Si un nom de cellule est indiqué par **Ref**, alors le mot clé **Sheet** n'est pas utile.

Sinon, si le mot clé **Sheet** est absent, c'est la feuille active qui est adressée.

Row, Col : Ces deux mots clés représentent respectivement :

- Les coordonnées (Ligne, Colonne) d'une cellule, si **Ref** est absent.
- Les coordonnées relatives en lignes et colonnes depuis la base référencée par "Ref=" quand ce dernier est présent. La cellule en haut à gauche a pour coordonnées relatives (1,1).

RowCnt, ColCnt : Ces deux mots clés représentent respectivement le nombre de lignes et de colonnes sélectionnées, à partir de la référence données par **Ref** ou par (**Row, Col**).

Ces mots clés ne peuvent pas être utilisés si **RefTo** est spécifié.

RefTo : Référence à la cellule en bas et à droite de la zone sélectionnée. Cette référence est donnée par des mots clé **Ref, Row et Col**, dans une chaîne entre guillemets.

Ce mot clé ne peut pas être spécifié si **RowCnt** ou **ColCnt** sont spécifiés.

NextOutline: Quand cette option est spécifiée, LAUNCHER Office recherche la prochaine ligne à partir de la cellule active, ayant un niveau de hiérarchie correspondant aux critères.

Les lignes ont des niveaux de hiérarchie différents quand des sous-totaux ont été appliqués à la feuille.

Dans la syntaxe **NextOutline="x-y"**, x représente le niveau de hiérarchie minimum, y le niveau maximum.

Exemple : La syntaxe suivante recherche la ligne de sous-total suivante et applique des formats couleurs de fond et de caractère :

```
NEXTOUTLINE="1-2"; SELECT; IgnoreError;  
PROP (INTERIOR.COLOR) = RGB (200;200;200) ; PROP (Font.Bold) = True
```

SetValue : Indique que le paramètre **Parm2** contient une valeur à affecter à la cellule.

SetFormula : Indique que le paramètre **Parm2** contient le texte d'une

formule à appliquer à la cellule.

Format : Permet de fixer le format de la cellule ou du groupe de cellules.

Name : Donne un nom à la cellule ou groupe de cellules.

Comment : Ajoute un commentaire à la cellule. Ce commentaire est affiché par Excel quand on survole la cellule avec la souris.

GetValue : Indique que l'on veut recevoir la valeur interne de la cellule, et non la valeur affichée. La valeur sera retournée dans le paramètre &RESULT.

GetText : Indique que l'on veut recevoir la valeur affichée de la cellule. La valeur affichée peut différer de la valeur interne. Par exemple, si la cellule a un format "Pourcentage", la valeur interne "0,10" sera affichée "10,00%". La valeur sera retournée dans le paramètre &RESULT.

GetProp("Propriété") : retourne la valeur d'une propriété affectée aux cellules de la région référencée.

Exemples :

Prop("Address") retourne les coordonnées de la région référencée.

Prop("Font.color") retourne la couleur des caractères.

GetSize : Retourne dans &RESULT la taille de la région référencée par **Ref**.

La taille est retournée dans &RESULT sous la forme :

lllll ;cccc

lllll représente le nombre de lignes sur 5 chiffres,

cccc représente le nombre de colonnes sur 5 chiffres.

Select : True indique que la cellule ou le groupe de cellules doivent être sélectionnés.

Clear : True efface la cellule.

Freeze : True remplace les expressions par leur valeur calculées.

Calculate : True recalcule l'ensemble du classeur.

Merge : Quand ce mot clé est spécifié, la fusion des cellules de la sélection est modifiée.

Merge peut prendre une des valeurs symboliques suivante, exprimée sans guillemets :

xlAll : L'ensemble des cellules de la sélection sont fusionnées en une seule cellule.

xlColumn : Les cellules de chaque ligne de la sélection sont fusionnées pour former une cellule fusionnée par ligne.

xlNone : La fusion est supprimée.

Attention, si des cellules fusionnées contiennent des valeurs, Excel affiche un message pour demander confirmation de la perte de ces valeurs.

Utilisez la commande XLSETPROP, avec "DisplayAlerts=false" pour désactiver l'affichage de ce message.

Prop() : permet de modifier une propriété des cellules pointées.

Le mot clé "**Prop**" peut figurer plusieurs fois dans Parm1.

Exemple : Changer la couleur de fond, et l'alignement

Prop(Interior.color)=RGB(100,0,0);Prop(HorizontalAlignment)=xlCenter.

IgnoreError : Si cette option est vraie, en cas d'erreur, aucun message sera généré, et la valeur « *ERROR » sera retournée dans &RESULT.

DateFmt(format) : Quand une valeur de date est fournie dans le paramètre "Parm2", ce mot clé indique dans quel format se trouve cette valeur.

Les chaînes **dd**, **mm**, **yy** et **yyyy** seront utilisées pour désigner respectivement, le jour, le mois, l'année sur 2 chiffres et l'année sur 4 chiffres.

Par défaut, Excel considère les dates au format '**mm/dd/yyyy**'

Exemples:

DateFmt(dd/mm/yyyy) indique que la date est fournie en format Jour-Mois-Année, séparés par '/'.
DateFmt(yyyymmdd), la date est au format Année-Mois-Jour, sans séparateur.

NumFmt(format) Quand un nombre décimal, sans séparateur décimal, est fourni dans le paramètre "Parm2", ce mot clé indique combien de décimales comporte ce nombre.

Exemple:

Parm2='123456';

NumFmt(2) indique que la valeur envoyée est : **1234,56**.

Unicode : Quand Unicode est vrai, la valeur lue par **GetValue**, **GetText**, ou écrite par **SetValue** est dans le jeu de caractère Unicode de Windows.

Pour écrire une valeur en Unicode, il faut que &PARAM2 contienne une chaîne Unicode, pouvant être convertie par le programme [LNCCVTWCS](#). Lors d'une lecture d'une valeur, le paramètre &RESULT contiendra une valeur en Unicode, que l'on pourra convertir avec [LNCCVTWCS](#).

Parm2

Si le mot clé **SetValue** ou **SetFormula** apparaissent dans Parm1, alors

| | |
|-----|---|
| | <p>Parm2 contient le texte de la valeur ou formule.</p> <p>Si l'option Unicode=True est fixée, le contenu de &Parm2 doit être en format Unicode Windows, et le paramètre &OPT doit avoir la valeur 'W'.</p> |
| Opt | <p>Quand Unicode est vrai, &OPT doit avoir la valeur 'W', et &PARM2 doit contenir une chaîne Unicode Windows.</p> |

Exemples

```

/* Lire la valeur de la cellule B2 */
/* et lui affecter une nouvelle valeur */
CHGVAR      VAR(&CMD) VALUE('XLCELLS')
CHGVAR      VAR(&PARM1) VALUE('Ref="$B$2";SetValue=True;GetText=True')
CHGVAR      VAR(&PARM2) VALUE(&NEWVAL)
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
/* &RESULT contient l'ancienne valeur de B2 */

```

```

/* Changer le format de la cellule Ligne 10, Colonne 5 */
CHGVAR      VAR(&CMD) VALUE('XLCELLS')
CHGVAR      VAR(&PARM1) VALUE('Row=10;Col=5;Format="#" #0,00"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

```

```

/* Ecrire un texte Chinois dans une cellule */
/* Convertir depuis CCSID chinois vers unicode Windows */
CHGVAR      VAR(&WCS_LEN) VALUE(512)
CHGVAR      VAR(&CCSID) VALUE(835)
CALL        PGM(LNCCVTWCS) PARM(&HANDLE '*TO ' &PARM2 &WCS_LEN +
&CHINA &CHINA_LEN &CCSID &RES_SIZE)

```

```

CHGVAR      VAR(&CMD) VALUE('XLCELLS')
CHGVAR      VAR(&PARM1) VALUE('Ref="$B$2";SetValue=True;Unicode=True')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD 'W' &PARM1 +
&PARM2 &RESULT)

```

```

/* Sélectionner une zone de 3 lignes, 5 colonnes */
/* A partir de la cellule nommée "Table". */
CHGVAR      VAR(&CMD) VALUE('XLCELLS')
CHGVAR      VAR(&PARM1) VALUE('Ref="Table";RowCnt=3;ColCnt=5;Select')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +

```

```
&PARM2 &RESULT)

/* Sélectionner une zone */
/* depuis la cellules nommée "Table" */
/* jusqu'à la cellule nommée "FinTable".*/
CHGVAR VAR(&CMD) VALUE('XLCELLS')
CHGVAR VAR(&PARM1) VALUE('Ref="Table";RefTo="FinTable";Select')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/* Sélectionner une zone */
/* depuis la cellules nommée "Table" */
/* jusqu'à la cellule Ligne 12 , colonne 20.*/
CHGVAR VAR(&CMD) VALUE('XLCELLS')
CHGVAR VAR(&PARM1) VALUE('Ref="Table";+
RefTo="Row=12;Col=20";Select')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/* Application formule */
LNCCMD CMD(XLCELLS) PARM1('Ref="H8";setformula=True;calculate=True')
PARM2('=IF(ISERROR(-1+D5/E5),"",-1+D5/E5)')
```

Voir aussi l'exemple : [Programme CL avec passage de paramètre](#)

Note

Pour utiliser la propriété **Font.Color**, il faut utiliser les codes couleur Windows.
Soit les constantes:

| Constant | Value | Description |
|------------------|----------|-------------|
| vbBlack | &H0 | Black |
| vbRed | &HFF | Red |
| vbGreen | &HFF00 | Green |
| vbYellow | &HFFFF | Yellow |
| vbBlue | &HFF0000 | Blue |
| vbMagenta | &HFF00FF | Magenta |
| vbCyan | &HFFFF00 | Cyan |
| vbWhite | &HFFFFFF | White |

Soit les valeurs:

| | |
|-----------|----------|
| vbBlack | 0 |
| vbBlue | 16711680 |
| vbCyan | 16776960 |
| vbGreen | 65280 |
| vbMagenta | 16711935 |
| vbRed | 255 |
| vbWhite | 16777215 |
| vbYellow | 65535 |

PROP(Font.Color)=vbMagenta
PROP(Font.Color)=INT(255) //pour rouge

On peut aussi utiliser les valeurs BGR avec la fonction RGB :
 RGB(blue,green,red).

PROP(Font.Color)=RGB(255,0,0) //bleu
 PROP(Font.Color)=RGB(0,255,0) // vert
 PROP(Font.Color)=RGB(0,0,255) // rouge

Commande *XLCLCLOSFILE*

Ferme le classeur EXCEL courant.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('XLCLCLOSFILE')
CHGVAR          VAR(&PARAM1) VALUE(['*YES|*NO'])
CHGVAR          VAR(&PARAM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Option de sauvegarde : *YES : sauvegarde les modifications et ferme le document. *NO : ferme le document sans sauvegarder les modifications apportées. Si aucune option n'est choisie, par défaut une fenêtre Office s'ouvrira pour le choix d'enregistrement: « Voulez-vous enregistrer les modifications apportées à classeur ? ». |

Note 1 :

Attention, il faut utiliser la commande suivante :

```
LNCCMD CMD(XLCLCLOSFILE) PARM1('*YES')
```

seulement sur un classeur qui a déjà été sauvegardé une fois.

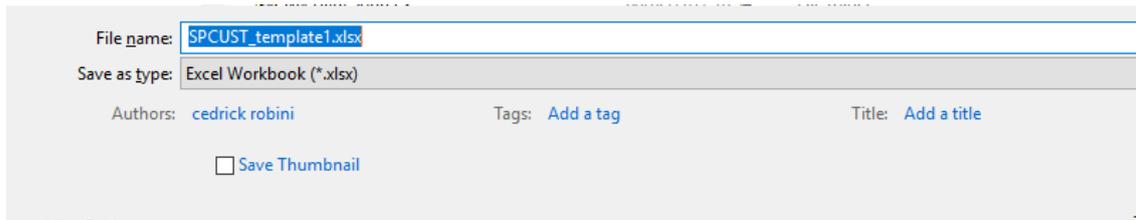
Il ne faut pas l'utiliser sur un nouveau classeur car sinon une pop-up Excel s'ouvrira (si Launcher est en interactif) pour demander de sauvegarder le classeur.

Exemple :

```
LNCTOXLS TOXLS('C:\A\Templates\SPCUST_template.xls')
FROMFILE(SP_CUST) TONAME(TABLEAU1) XLSMAP(*MAPNAME) SAVFMT(*NORMAL)
ENDOPT(*NONE) SAVDOC(*NONE)
```

```
LNCCMD CMD(XLCELLS) PARM1('Ref="$C$11";Setvalue=true') PARM2('1')
LNCCMD CMD(XLCLCLOSFILE) PARM1('*YES')
```

après un LNCTOXLS ENDOPT(*NONE) SAVDOC(*NONE), un nouveau classeur basé sur le modèle est créé. Après XLCLCLOSFILE *YES, on a cette pop-up Excel qui demande l'enregistrement du document:



Note 2 :

Attention si vous travaillez en mode compatibilité avec une version d'Excel supérieure à 2003 !
Par défaut le paramètre SAVFMT de LNCTOXLS vaut *NORMAL, c'est-à-dire le format par défaut d'Excel. Avec les versions supérieures à 2003 , le format par défaut d'Excel est : XLSX

Exemple :

```
LNCTOXLS TOXLS('C:\A\Templates\SPCUST_template.xls')
FROMFILE(SP_CUST) TONAME(TABLEAU1) XLSMAP(*MAPNAME)
SAVDOC('c:\temp\res11233.xls') SAVFMT(*NORMAL) ENDOPT(*NONE)

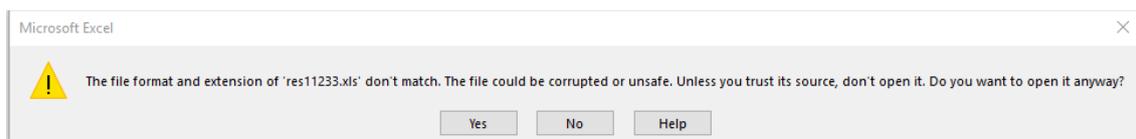
LNCCMD CMD(XLCELLS) PARM1('Ref="$C$11";Setvalue=true') PARM2('1')

LNCCMD CMD(XLCLOSFILE) PARM1('*YES')

LNCCMD CMD(EXCELCLOSE)
```

Le classeur généré ne sera pas correct : l'extension (.xls) et le format ne seront pas homogènes. En effet le format par défaut d'Excel 2016 est .XLSX.

Voici la pop-up Excel lors de l'ouverture du classeur généré :



La bonne syntaxe est la suivante :

```
LNCTOXLS TOXLS('C:\A\Templates\SPCUST_template.xls')
FROMFILE(SP_CUST) TONAME(TABLEAU1) XLSMAP(*MAPNAME)
SAVDOC('c:\temp\res11233.xls') SAVFMT(*XLS) ENDOPT(*NONE)

LNCCMD CMD(XLCELLS) PARM1('Ref="$C$11";Setvalue=true') PARM2('1')

LNCCMD CMD(XLCLOSFILE) PARM1('*YES')

LNCCMD CMD(EXCELCLOSE)
```

Voir aussi

- [XLOPENFILE](#)
- [XLSAVEAS](#)

Commande *XLCOPY*

Copie la sélection courante ou le contenu d'une cellule dans le presse-papier ou dans une cellule spécifiée.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLCOPY')
CHGVAR          VAR(&PARM1) VALUE('
                [From="Coordonnées Cellules à copier"];
                [To="Coordonnées Cellule de destination"]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>From : Désigne les coordonnées des cellules dont le contenu doit être copié. Si From n'est pas précisé, c'est la sélection courante qui est copiée vers le presse-papier.</p> <p>To : Désigne les coordonnées des cellules vers lesquelles doit se faire la copie. Si To n'est pas précisé, les données sont copiées dans le presse-papier.</p> |

Remarques

1)

Les cellules peuvent se situer dans la feuille courante ou dans une autre feuille du classeur. Le nom de la feuille et les coordonnées de la cellule seront alors séparés par un point d'interrogation.

Exemple : dest?F3 pour la cellule F3 de la feuille nommée "dest".

2)

Il faut mettre le nom de la feuille entre guillemets si celui-ci contient des espaces.

Exemple : "dest feuille"?A1 pour la cellule A1 de la feuille nommée "dest feuille".

Exemples

L'exemple suivant copie le contenu de "Revenus" à l'emplacement "\$B\$22".

```
CHGVAR      VAR(&CMD) VALUE('XLCOPY)
CHGVAR      VAR(&PARM1) VALUE('From="Revenus";To="$B$22"')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

L'exemple suivant copie la sélection courante vers le presse papier.

```
CHGVAR      VAR(&CMD) VALUE('XLCOPY)
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Copie du contenu de la zone A1:B2 de la feuille "or feuille" dans la feuille "dest".
La cellule en haut à gauche de la zone de réception est la cellule F3.

```
LNCCMD CMD(XLCOPY) PARM1('From=""or feuille"?A1:B2";To="dest?F3"')
```

Voir aussi

- [XLPASTE](#)

Commande *XLCOPYSH*

Copie la feuille courante dans le même classeur. La copie apparait après la feuille copiée.

La copie devient alors la feuille courante.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('XLCOPYSH')
CHGVAR          VAR(&PARAM1) VALUE(' ')
CHGVAR          VAR(&PARAM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Exemple

```
CHGVAR VAR(&CMD) VALUE('XLOPENFILE')
CHGVAR VAR(&PARAM1) VALUE('FILE="C:\TEMP\CLASSEUR1.XLSX"')
CHGVAR VAR(&PARAM2) VALUE('VISIBLE')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)

CHGVAR VAR(&CMD) VALUE('XLCOPYSH')
CHGVAR VAR(&PARAM1) VALUE(' ')
CHGVAR VAR(&PARAM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Commande **XLDELSHEET**

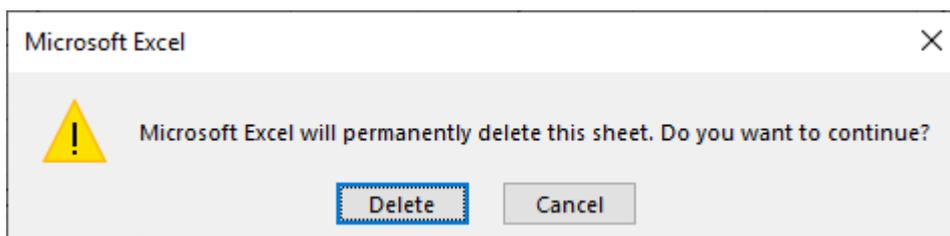
Supprime la feuille courante. Les feuilles à la droite de celle supprimée sont décalées à gauche.

Syntaxe

```
CHGVAR      VAR (&CMD)  VALUE ('XLDELSHEET')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                               &PARM2 &RESULT)
```

Remarque

Quand on veut supprimer une feuille Excel avec l'instruction **XLDELSHEET**, le système envoie une fenêtre pour confirmation :



Une solution pour éviter cette fenêtre de Excel, est de fixer la propriété "DisplayAlerts" à "False".

Utiliser LNCCMD avec la commande **XLSETPROP** :

```
CHGVAR VAR (&CMD)  VALUE (XLSETPROP)
CHGVAR VAR (&PARM1) VALUE ('displayalerts')
CHGVAR VAR (PARM2) VALUE (false)
CALL   PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                           &PARM2 &RESULT)
```

Voir aussi

- [XLADDSHEET](#)
- [XLGOTOSH](#)
- [XLPRINTSH](#)
- [XLSHNAME](#)
- [XLSETPROP](#)

Commande *XLDISADD*

Désactive un complément EXCEL (activé dans la liste : File, Options, Add-ins, Excel add-ins, Go).

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('XLDISADD')
CHGVAR          VAR (&PARM1) VALUE ('Complément')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Chemin complet vers le complément à désactiver. |
| Parm2 | |

Exemple

```
LNCCMD          CMD(XLDISADD) PARM1('C:\A\ModelMacro2.xlam')
```

Voir aussi

- [XLADDINS](#)
- [XLENAADD](#)

Commande *XLDPROTECT*

Retire la protection contre les modifications appliquée à la feuille courante.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLDPROTECT')
CHGVAR      VAR(&PARM1) VALUE('MotDePasse')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Mot de passe de protection de la feuille courante. |
| Parm2 | |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('XLDPROTECT')
CHGVAR      VAR(&PARM1) VALUE('Aura')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLPROTECT](#)

Commande XLDRAWGR

Crée un graphique sur une nouvelle feuille.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLDRAWGR')
CHGVAR      VAR(&PARM1) VALUE('sélection')
CHGVAR      VAR(&PARM2) VALUE(' 'TYPE' '[ROWS|COLS]' +
                              'Nb_Categorie' 'Nb_Serie' 'Name' )
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                              &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Sélection des cellules de données contenant les données et/ou les étiquettes de données (coordonnées séparées par un : - deux points -). Si plusieurs cellules sont indiquées, elles doivent être séparées par un ; (point-virgule).</p> <p>Le nombre de caractères utilisés pour définir la sélection ne doit pas dépasser 260. Il faut utiliser des zones de cellules pour limiter le nombre de caractères.</p> |
| Parm2 | <p>Autres paramètres</p> <ul style="list-style-type: none"> - Le type de graphique (AREA, BAR, COLUMN, LINE, RADAR, DONUT, XYSCATTER, 2DPIE, 3DAREA, 3DBAR, 3DCOLUMN, 3DLINE, 3DPIE). - La disposition des données (ROWS ou COLS). - Le nombre de lignes ou de colonnes de la plage source qui contiennent les étiquettes des catégories. - Le nombre de lignes ou de colonnes de la plage source qui contiennent les étiquettes des séries. - Le titre du graphique. |

Exemple 1

Construire un graphique d'après le tableau Excel suivant :

| | A | B | C |
|---|---------------|-------------------|-------------------|
| 1 | Années | ITEMSTOTAL | AMOUNTPAID |
| 2 | 2001 | 162 730,05 € | 162 730,05 |
| 3 | 2002 | 315 910,20 € | 273 096,75 |
| 4 | 2003 | 162 379,40 € | 162 379,40 |



| | | | |
|---|------|--------------|------------|
| 5 | 2004 | 953 405,35 € | 786 008,75 |
| 6 | 2005 | 967 225,30 € | 935 238,30 |
| 7 | 2006 | 360 950,80 € | |
| 8 | 2007 | 65,00 € | |

PGM

```
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL VAR(&PARAM1) TYPE(*CHAR) LEN(512)
DCL VAR(&PARAM2) TYPE(*CHAR) LEN(1024)
DCL VAR(&RESULT) TYPE(*CHAR) LEN(512)
```

LNCOPEN

LNCCMD CMD(EXCELOPEN)

```
LNCCMD CMD(XLOPENFILE) +
  PARM1('%LNCDIR%\samples\test.xls') +
  PARM2(visible)
```

```
LNCCMD CMD(XLDRAWGR) PARM1('A1:C8') +
  PARM2(''COLUMN'' 'COLS'' '1'' '1'' 'TEST1'')
```

LNCCMD CMD(EXCELSHOW)

LNCCMD CMD(END)

```
SNDPGMSG MSGID(CPF9898) MSGF(QCPFMSG) +
  MSGDTA(&RESULT) MSGTYPE(*INFO)
```

LNCCLOSE

ENDPGM

Exemple 2

PGM

LNCOPEN

LNCCMD CMD(EXCELOPEN)

```
LNCCMD CMD(XLOPENFILE) +
  PARM1('%LNCDIR%\samples\test.xlsx')
```

```
LNCCMD CMD(XLDRAWGR) PARM1('Sheet1!$A$1:C$4') +
  PARM2(''COLUMN'' 'COLS'' '1'' '1'' +
  'TEST1'')
```

```
LNCCMD CMD(XLMETHOD) +
  PARM1('Sheets.Item("Chart1").Select')
```

LNCCMD CMD(EXCELSHOW)



```
LNCCLOSE
ENDPGM
```

Exemple 3: plusieurs cellules définies (limite 260 caractères)

```
LNCCMD      CMD (XLDRAWGR) +
             PARM1 ('Sheet1!$A$1;Sheet1!$A$2;Sheet1!$A$3;+
             Sheet1!$A$4;Sheet1!$A$5;Sheet1!$A$6;Sheet1!+
             $A$7;Sheet1!$A$8;Sheet1!$B$1;Sheet1!$B$2;Sh+
             eet1!$B$3;Sheet1!$B$4;Sheet1!$B$5;Sheet1!$B+
             $6;Sheet1!$B$7;Sheet1!$B$8;Sheet1!$C1;Sheet+
             1!$C$2;Sheet1!$C$3') PARM2 (''COLUMN'' +
             ''COLS'' ''1'' ''1'' ''TEST1''')
```

Exemple 4: utilisation zone de données "data"

```
LNCCMD      CMD (XLDRAWGR) PARM1 ('Sheet1!data') +
             PARM2 (''COLUMN'' ''COLS'' ''1'' ''1'' +
             ''TEST1''')
```

Exemple 5 :

```
CHGVAR      VAR (&CMD) VALUE ('XLDRAWGR')
CHGVAR      VAR (&PARM1) VALUE (' A2:D3 ')
CHGVAR      VAR (&PARM2) +
             VALUE (' ''3DPIE'' ''ROWS'' ''1'' ''0'' ''Market''
             ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
             &PARM1 &PARM2 &RESULT)
```

Notes

Les fonctions de tracé de graphiques dans LAUNCHER sont limitées avec **XLDRAWGR**.

Pour modifier certaines propriétés du graphique, vous pouvez utiliser la commande [XLMETHOD](#) qui permet d'exécuter depuis l'AS/400 des méthodes VBA Excel :

```
LNCCMD CMD (XLMETHOD) PARM1 ('ActiveSheet.ChartObjects.Item("Graphique
1").Activate')
LNCCMD CMD (XLMETHOD)
PARM1 ('ActiveChart.ApplyDataLabels (xlDataLabelsShowValue)')
```

Pour avoir une idée des méthodes à appeler, placez vous sous Excel en mode "Enregistrement de macros" : Menu **"Outils" – "Macro" – "Nouvelle macro"**.

Faites les opérations désirées au clavier et souris.

Arrêtez l'enregistrement de macro, et allez voir le code généré par Excel : Menu **"Outils" – "Macro" – "Macros" – "Modifier"**

Exemple :

La macro qui active le graphique et modifie une propriété est la suivante :

```
ActiveSheet.ChartObjects("Graphique 1").Activate  
ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowPercent,  
LegendKey:=True _  
, HasLeaderLines:= False
```

Note : Il y a 2 différences à noter entre la syntaxe VB de Excel et la syntaxe LAUNCHER :

1) Quand on sélectionne un objet d'une collection, il faut ajouter ".Item" après le nom de la collection.

Pour sélectionner "Graphique 1" dans la collection des graphiques, la syntaxe VB est la suivante :

```
ActiveSheet.ChartObjects("Graphique 1")
```

Elle devient :

```
ActiveSheet.ChartObjects.Item("Graphique 1")
```

2) Les noms de paramètres (Nom:=valeur) est propre à la syntaxe VB.

Avec LAUNCHER, il faut lister les valeurs de chaque paramètre, dans l'ordre attendu, séparés par ';'.

L'instruction suivante :

```
ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowPercent,  
LegendKey:=True
```

Devient :

```
ActiveChart.ApplyDataLabels (xlDataLabelsShowPercent;True)
```

Sinon, optez pour l'écriture d'une macro sous Excel, que vous appellerez par [XLEXEMACRO](#).

Commande *XLENAADD*

Active un complément EXCEL disponible dans la liste (File, Options, Add-ins, Excel add-ins, Go).

Un complément peut être également ajouté à la liste et activé avec la commande [XLADDINS](#).

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('XLENAADD')
CHGVAR      VAR (&PARM1) VALUE ('Complément')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin complet vers le complément à activer. |
| Parm2 | |

Exemple

```
LNCCMD      CMD (XLENAADD) PARM1 ('C:\A\ModelMacro2.xlam')
```

Voir aussi

- [XLADDINS](#)
- [XLDISADD](#)

Commande XLEXEMACRO

Exécute une macro EXCEL.

La macro peut être présente dans le classeur ouvert (.xls ou .xlsm par exemple), ou bien dans un complément Excel (.xla ou .xlam), que l'on pourra charger avec la commande [XLADDINS](#).

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLEXEMACRO')
CHGVAR      VAR(&PARM1) VALUE('Nom de la macro')
CHGVAR      VAR(&PARM2) VALUE('[Paramètres macro]')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Nom de la macro. |
| Parm2 | Paramètres passés à la macro. Les paramètres doivent être entre guillemets, séparés par un point virgule. |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('XLEXEMACRO')
CHGVAR      VAR(&PARM1) VALUE('Macro1')
CHGVAR      VAR(&PARM2) VALUE('"12000";"FRANCE" ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

ou

```
CHGVAR      VAR(&CMD) VALUE('XLEXEMACRO')
CHGVAR      VAR(&PARM2) VALUE('Macro')
CHGVAR      VAR(&PARM1) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT))
```

Dans cet exemple, le classeur actif contient les macros.

Autre exemple

Supposons que nous ayons la macro suivante, avec deux paramètres :

```
Sub ProcMsgBox(sParam1 As String, sParam2 As String)

MsgBox sParam1 & " - " & sParam2

End Sub
```

Si la macro est dans le classeur actif, alors on utilisera ce code pour exécuter la macro :

```
LNCCMD CMD(EXCELOPEN)
LNCCMD CMD(XLOPENFILE) PARM1('C:\A\temp.xlsm')
LNCCMD CMD(XLEXEMACRO) PARM1('ProcMsgBox') +
PARM2('"Aura";"Equipements"')
```

Si la macro est dans un complément XLAM (ou un complément XLA), alors on utilisera le code suivant pour charger le complément et ensuite exécuter la macro :

```
LNCCMD CMD(EXCELOPEN)
LNCCMD CMD(XLOPENFILE) PARM1('C:\A\temp.xlsx')
LNCCMD CMD(XLADDINS) PARM1('C:\A\ModelMacro.xlam')
LNCCMD CMD(XLEXEMACRO) PARM1('ProcMsgBox') +
PARM2('"Aura";"Equipements"')
```

Voir aussi

- [XLMETHOD](#)
- [XLADDINS](#)

Commande XLEXISTSH

Vérifie l'existence d'une feuille dans le classeur courant.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLEXISTSH')
CHGVAR      VAR(&PARM1) VALUE('NomFeuille')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Le nom de la feuille à rechercher. |
| RESULT | Si la feuille existe, la valeur TRUE est retournée. Sinon, c'est la valeur FALSE . Le résultat transite par le message de retour provenant du PC (variable &RESULT). |

Exemple

```
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)

LNCOOPEN

LNCCMD      CMD(EXCELOPEN)

LNCCMD      CMD(XLOPENFILE) PARM1('C:\A\temp.xlsx')

LNCCMD      CMD(EXCELSHOW)
LNCCMDR     CMD(XLEXISTSH) PARM1('tab') RESULT(&RESULT)
LNCCMD      CMD(NOP) PARM1(&RESULT)
```

Voir aussi

- [CHKFILE](#)

Commande *XLFIN*D

Recherche une valeur dans la feuille courante.

La cellule qui contient la valeur recherchée est activée.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLFIN'D')
CHGVAR      VAR(&PARM1) VALUE('valeur')
CHGVAR      VAR(&PARM21) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | La valeur à rechercher dans la feuille. |
| RESULT | Si la valeur est trouvée, la valeur TRUE est retournée. Sinon, c'est la valeur FALSE . Le résultat transite par le message de retour provenant du PC (variable &RESULT). |

Exemple

```
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)

LNCOPEN

LNCCMD      CMD(EXCELOPEN)

LNCCMD      CMD(XLOPENFILE) PARM1('C:\A\temp.xlsx')

LNCCMD      CMD(EXCELSHOW)
LNCCMDR     CMD(XLFIN'D) PARM1('Aura') RESULT(&RESULT)
LNCCMD      CMD(NOP) PARM1(&RESULT)
```

Voir aussi

- [XLREPLACE](#)

Commande *XLFORMULA*

Ecrit une formule dans une cellule d'un classeur Excel.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLFORMULA')
CHGVAR      VAR(&PARM1) VALUE('[coordonnées cellule]')
CHGVAR      VAR(&PARM2) VALUE('formule')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Coordonnées de la cellule à modifier. Facultatif. Si Parm1 n'est pas renseigné, la cellule active sera prise en compte.</p> <p>Les coordonnées de la cellule peuvent être données sous la forme :</p> <ul style="list-style-type: none"> - Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3. - Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B). - Nom symbolique donné à une cellule. |
| Parm2 | <p>Formule à exécuter.</p> |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('XLFORMULA')
CHGVAR      VAR(&PARM1) VALUE('$A$2')
CHGVAR      VAR(&PARM2) VALUE('=B8+E10')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

ou

```
CHGVAR      VAR(&CMD) VALUE('XLFORMULA')
CHGVAR      VAR(&PARM1) VALUE('A2')
CHGVAR      VAR(&PARM2) VALUE('=B8+E10')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```



Commande XLGETFILE

Importe un fichier texte dans une feuille d'un classeur Excel.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('XLGETFILE')
CHGVAR      VAR(&PARM1) VALUE('fichier texte')
CHGVAR      VAR(&PARM2) VALUE('
[Start=Numéro]
[;RecordCnt=Numéro]
[;Destination=Référence cellule]
[;MapColName=True/False]
[;AutoFit=True/False]
[;FormatValue=True/False]
')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin et nom du fichier texte à importer dans la feuille Excel. |
| Parm2 | <p>Parm2 permet de fixer des propriétés de lecture du fichier texte.</p> <p>Les propriétés sont séparées par des points virgules. Le nom des propriétés peut être exprimé en lettres minuscules ou majuscules.</p> <p><u>Propriétés :</u></p> <p>Start : pour fixer le rang de la première ligne du fichier source à insérer. Si le fichier source est issue d'un transfert effectué par la commande DBXFER, alors il inclut les noms des colonnes en première ligne. Cette ligne sera ignorée lors de l'insertion avec "Start=2".</p> <p>RecordCnt : pour fixer le nombre de lignes maximal à insérer.</p> <p>Destination : permet de donner la référence de la cellule en haut à gauche de la plage de destination qui recevra les données du fichier. Par défaut, c'est la cellule active qui sera la première cellule de réception. Ce paramètre n'est pas pris en compte si MapColName = true.</p> <p>Si MapColName est vrai, les cellules portant les noms des colonnes du fichier texte reçoivent les valeurs.</p> |



La première ligne du fichier texte doit contenir les noms des colonnes.

Si **Autofit** est vrai, cela permet d'ajuster la largeur des colonnes au contenu.

FormatValue :

True (par défaut) : Le format des cellules est fixé en fonction du type des données AS400.

False : Le format des cellules est fixé par Excel, en fonction des valeurs.

Exemple

```
LNCCMD      CMD (XLOPENFILE) +
             PARM1 ('C:\A\Templates\SPCUST_template.xlsx')

LNCCMD      CMD (XLGOTOSH) PARM1 ('Invoice')

LNCCMD      CMD (XLGETFILE) +
             PARM1 ('File="C:\A\LNC002.TXT"') +
             PARM2 ('Start=2;FormatValue=True;Aut+
             oFit=False;RecordCnt=4;MapColName=true')

LNCCMD      CMD (XLGOTOSH) PARM1 ('Data')

LNCCMD      CMD (XLGETFILE) +
             PARM1 ('File="C:\A\LNC002.TXT"') +
             PARM2 ('Start=2;FormatValue=True;Aut+
             oFit=true;RecordCnt=4;Destination=$B$3')
```

Dans cet exemple, les données du fichier texte (4 enregistrements), à partir du deuxième enregistrement, sont copiées dans la feuille "Invoice", en utilisant le mappage des données en fonction de leurs noms.

De plus les données du fichier texte (4 enregistrements), à partir du deuxième enregistrement, sont aussi copiées dans la feuille "Data". La cellule en haut à gauche de la plage de destination est la cellule B3.

Voir aussi

- [LNCTOXLS - Commande](#)

Fichier de description (*.LFD)

La commande XLGETFILE prend en compte le fichier LFD, s'il existe.

Ce fichier porte le même nom que le fichier texte à importer, mais avec le suffixe "LFD".

La première ligne du fichier LFD contient le nom du fichier texte à importer, sans son extension, entre crochets [].

[Mon Fichier]

Les lignes suivantes décrivent le format de chaque colonne du fichier texte à importer. Il y a une ligne dans le fichier LFD par colonne du fichier texte.

Format d'une ligne de description : **N=Type,format**

N = Numéro de colonne. La première colonne porte le numéro 0.

Type =

Num : Pour une colonne de type numérique.

Format : **n.d**

n=Nombre de chiffres

d=Nombre de décimales

Text : Pour une colonne de type texte.

Format : **n**

n est le nombre de caractères (optionnel)

Date : Pour une colonne de type date.

Format :

DMY : Jour/Mois/Année

MDY : Mois/Jour/Année

YMD : Année/Mois/Jour

Skip : Pour ignorer la colonne, et ne pas l'importer.

Exemple :

```
[LNC002]
0=Num,5.0
1=Num,5.0
2=Date,YMD
3=Date,YMD
4=Num,5.0
5=Text,7
6=Skip
7=Text,7
8=Num,8.2
```

Commande *XLGETPOS*

Retourne la position courante du curseur dans la feuille active.

La valeur retournée est de la forme **\$XX\$YYYY**.

XX étant le numéro de la colonne (la lettre).

YYYY étant le numéro de la ligne.

Le résultat transite par le message de retour provenant du PC (variable &RESULT).

Syntaxe

```

CHGVAR          VAR (&CMD) VALUE ('XLGETPOS')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
    
```

Exemple

Supposons que la cellule suivante soit activée :

| | A | B | C | D |
|---|---|-----|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | fff | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |

Après l'exécution de la commande suivante :

```
LNCCMD CMD (XLGETPOS)
```

la variable &RESULT contiendra la valeur suivante : \$B\$3 .

Commande XLGETPROP

Lit la valeur d'une propriété sur le classeur, la feuille, une cellule ou une sélection.

Syntaxe

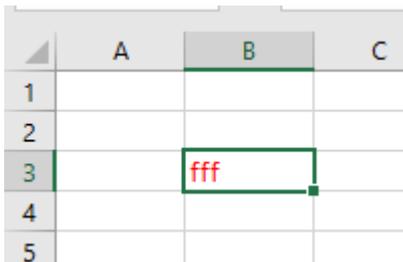
```
CHGVAR      VAR(&CMD) VALUE('XLGETPROP')
CHGVAR      VAR(&PARM1) VALUE('Propriété')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Le chemin complet vers la propriété à lire. Exemple : ActiveCell.Font.Color |
| RESULT | Reçoit la valeur de la propriété après l'appel. |

Exemple

Supposons que la cellule suivante soit activée :



Après l'exécution de la commande suivante :

```
CHGVAR      VAR(&CMD) VALUE('XLGETPROP')
CHGVAR      VAR(&PARM2) VALUE('ActiveCell.Font.Color')
CHGVAR      VAR(&PARM1) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

la variable &RESULT contiendra la valeur suivante : 255, qui correspond au code couleur Rouge.

Pour rappel, voici les codes couleur Windows :

| Constant | Value | Description |
|------------------|----------|-------------|
| vbBlack | &H0 | Black |
| vbRed | &HFF | Red |
| vbGreen | &HFF00 | Green |
| vbYellow | &HFFFF | Yellow |
| vbBlue | &HFF0000 | Blue |
| vbMagenta | &HFF00FF | Magenta |
| vbCyan | &HFFFF00 | Cyan |
| vbWhite | &HFFFFFF | White |

Voir aussi

- [XLSETPROP](#)

Commande *XLGETVALUE*

Récupère la valeur d'une cellule de la feuille courante.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLGETVALUE')
CHGVAR      VAR(&PARM1) VALUE('Coordonnées')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Les coordonnées de la cellule dont on veut récupérer la valeur.</p> <p>Les coordonnées peuvent être données sous la forme :</p> <ul style="list-style-type: none"> - Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3. - Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B). - Nom symbolique donné à une cellule ou un groupe de cellules. <p>Si les coordonnées ne sont pas renseignées, c'est la cellule active qui est prise en compte.</p> |
| RESULT | Le résultat transite par le message de retour provenant du PC (variable RESULT). |

Exemple

Pour récupérer la valeur de la cellule nommée "Total" :

```
CHGVAR      VAR(&CMD) VALUE('XLGETVALUE')
CHGVAR      VAR(&PARM1) VALUE('Total')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)

/ * POUR AFFICHER LE RESULTAT A L'ECRAN */
SNDFGMMMSG  MSGID(CPF9898) MSGF(QSYS/QCPFMSG) + MSGDTA(&RESULT)
TOPGMQ(*EXT) MSGTYPE(*STATUS)
```

Pour récupérer la valeur de la cellule C4 :

```
LNCCMD CMD(XLGETVALUE) PARM1('C4')
```

Voir aussi

- [XLSETTEXT](#)
- [XLSETVALUE](#)

Commande *XLGOTOCELL*

Déplace la position du curseur dans la feuille courante.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLGOTOCELL')
CHGVAR      VAR(&PARM1) VALUE('coordonnées')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Les coordonnées de la cellule de destination. Les coordonnées peuvent être données sous la forme : - Syntaxe \$B\$3 . Exemple : \$B\$3 désigne la colonne B, ligne 3. - Nom symbolique donné à une cellule ou un groupe de cellules. |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('XLGOTOCELL')
CHGVAR      VAR(&PARM2) VALUE('Total')
CHGVAR      VAR(&PARM1) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

ou

```
CHGVAR      VAR(&CMD) VALUE('XLGOTOCELL')
CHGVAR      VAR(&PARM2) VALUE('$B$3')
CHGVAR      VAR(&PARM1) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLCELLNAME](#)
- [XLSETTEXT](#)
- [XLSETVALUE](#)

Commande *XLGOTOSH*

Active une feuille du classeur courant.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('XLGOTOSH')
CHGVAR      VAR (&PARAM1) VALUE ('nom')
CHGVAR      VAR (&PARAM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Le nom de la feuille à activer. Le nom peut être celui fixé par l'appel à la commande XLSHNAME . |
| Parm2 | |

Exemple

```
CHGVAR      VAR (&CMD) VALUE ('XLGOTOSH')
CHGVAR      VAR (&PARAM2) VALUE ('Data')
CHGVAR      VAR (&PARAM1) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Voir aussi

- [XLADDSHEET](#)
- [XLDELSHEET](#)
- [XLPRINTSH](#)
- [XLSHNAME](#)

Commande *XLINSERT*

Insère une cellule, une plage de cellules, une ligne, une colonne dans la feuille de travail et décale les autres cellules.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLINSERT')
CHGVAR          VAR(&PARM1) VALUE('
[Ref="Référence de cellules"]
[;Sheet="Nom de la feuille"]
[;Row=Numéro de ligne ou déplacement]
[;Col=Numéro de colonne ou déplacement]
[;Shift=xlShiftDown/xlShiftToRight]
[;Count=nombre]
[;EntireRow=True/False]
[;EntireColumn=True/False]
[;After=True/False]
[;Fill=True/False]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 + &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Ref : La référence à la cellule peut être donnée sous la forme :</p> <ul style="list-style-type: none"> - Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3. - Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B). - Nom symbolique donné à une cellule ou un groupe de cellules. - "." désignera la cellule active. <p>Sheet : Indique le nom de la feuille à adresser. Si un nom de cellule est indiqué par Ref, alors le mot clé Sheet n'est pas utile. Sinon, si le mot clé Sheet est absent, c'est la feuille active qui est adressée.</p> <p>Row, Col : Ces deux mots clés représentent respectivement :</p> <ul style="list-style-type: none"> - Les coordonnées (Ligne, Colonne) d'une cellule, si Ref est absent. - Les coordonnées relatives en lignes et colonnes depuis la base référencée par "Ref=" quand ce dernier est présent. La cellule en haut à gauche a pour coordonnées relatives (1,1). <p>Shift= <i>xlShiftToRight</i> pour décaler les cellules vers la droite, ou <i>xlShiftDown</i> (<u>défaut</u>) pour décaler les cellules vers le bas.</p> |

Count=Nombre d'insertions à effectuer.

EntireRow=*True* pour insérer des lignes entières, ou *False* (*défaut*) pour décaler les cellules vers le bas sur la largeur de la sélection. Cette option n'est valide que si **Shift**= *xlShiftDown*.

EntireColumn= *True* pour insérer des colonnes entières, ou *False* (*défaut*) pour décaler les cellules vers la droite sur la hauteur de la sélection. Cette option n'est valide que si **Shift**= *xlShiftToRight*.

After=*True* pour insérer après la sélection, *False* (*défaut*) pour insérer à partir de la sélection.

Fill=*True* pour copier la sélection sur l'ensemble des nouvelles cellules insérées, *False* (*défaut*) pour laisser les nouvelles cellules vides. Attention, quand **Fill** est vrai, il faut aussi que **After** soit vrai. Cette option permet de copier les formules de calculs de la première ligne d'un tableau, vers toutes les nouvelles lignes.

Exemple

L'exemple suivant insère 1 ligne entière, à la position de la cellule active courante.

```
CHGVAR          VAR(&CMD) VALUE('XLINSERT')
CHGVAR          VAR(&PARAM1) VALUE('EntireRow=True')
CHGVAR          VAR(&PARAM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Insère 10 lignes entières, après la ligne 5, et dupliquer la ligne 5 sur les 10 lignes insérées.

```
CHGVAR          VAR(&CMD) VALUE('XLINSERT')
CHGVAR          VAR(&PARAM1) +
VALUE('Ref="$A$5";Count=10;EntireRow=True; +
After =true;Fill=True')
CHGVAR          VAR(&PARAM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
&PARAM2 &RESULT)
```

Voir aussi

- [XLCCELLS](#)

Commande *XLINSERTCL*

Insère une colonne à gauche de la cellule active dans la feuille courante.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLINSERTCL')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLINSERTRW](#)

Commande *XLINSERTRW*

Insère une ligne au dessus de la cellule active dans la feuille courante.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('XLINSERTRW')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLINSERTCL](#)

Commande *XLITALIC*

Passer le contenu de la cellule active en caractères italiques.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLITALIC')
CHGVAR      VAR(&PARM2) VALUE(' ')
CHGVAR      VAR(&PARM1) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLBOLD](#)
- [XLUNDERLN](#)

Commande XLMAXIMIZE

Permet de maximiser la fenêtre EXCEL courante.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('XLMAXIMIZE')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLMINIMIZE](#)

Commande *XMLMENU*

Construit une barre de menu personnalisée sous EXCEL.

XMLMENU doit être utilisé avec les commandes **XLADDINS** et **XLWAIT**.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('XMLMENU')
CHGVAR      VAR(&PARM1) VALUE('
Captions="Libellés des boutons"
[;Tips="Bulle d'aide"]
[;Remove]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Captions : Permet de donner le libellé de chaque bouton de la nouvelle barre de menu. Les libellés sont séparés entre eux par un point virgule (;). Le nombre de libellés détermine le nombre de boutons de la barre de menu.</p> <p>Tips : Permet d'afficher une bulle d'aide pour chaque bouton. Les tips sont séparés entre eux par un point virgule (;). Leur nombre doit correspondre au nombre de libellés.</p> <p>Remove : supprime le précédent menu.</p> |

Remarques

1)

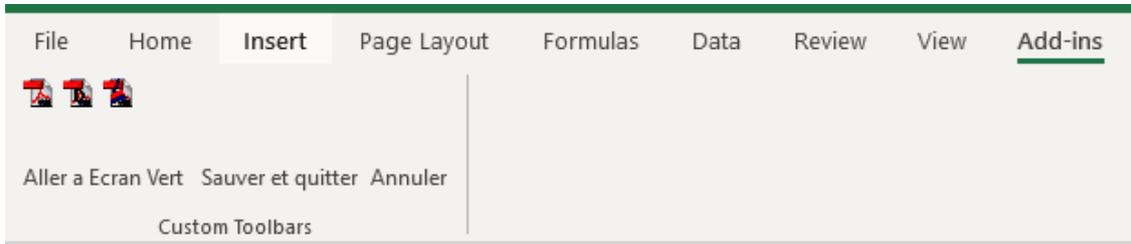
Le complément "**LNCEExcelAddin.xla**" doit être chargé par la commande **XLADDINS** avant d'utiliser **XMLMENU**.

Lors de l'appel de la commande **XLWAIT**, le programme AS/400 se met en attente d'une action de la part de l'utilisateur. Il reprend la main quand Excel est fermé, ou sur l'action d'un des boutons du nouveau menu.

Au retour de la commande **XLWAIT**, la variable &RESULT contient le numéro sur 5 chiffres du bouton actionné (de 1 à n).

2)

Le menu personnalisé apparaîtra dans l'onglet "Complément" du menu Excel.



Exemple

```
CHGVAR      VAR (&CMD) VALUE ('XLADDINS')
CHGVAR      VAR (&PARM1) VALUE ('%LNCDIR%\LNCEXcelAddin.xla')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR      VAR (&CMD) VALUE ('XLMENU')
CHGVAR      VAR (&PARM1) VALUE ('Captions="Valider;Annuler";Tips="Valider le
document;Abandonner le document"')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

```
CHGVAR      VAR (&CMD) VALUE ('XLWAIT')
CHGVAR      VAR (&PARM1) VALUE (' ')
CHGVAR      VAR (&PARM2) VALUE (' ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande *XMLMETHOD*

Appelle une méthode de l'objet Excel.Application.

Syntaxe

```

CHGVAR          VAR (&CMD) VALUE ('XMLMETHOD')
CHGVAR          VAR (&PARM1) VALUE ('Méthode')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG         MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Le chemin complet de la méthode à appeler. |

Remarque

La commande **XMLMETHOD** permet d'exécuter depuis l'AS/400 des méthodes du langage VBA Excel.

Le contenu de Parm1 respecte la syntaxe utilisée en Visual Basic.

Il est possible d'utiliser toutes les constantes d'Excel et Visual Basic Application ou bien la valeur de la constante elle-même.

Pour avoir une idée des méthodes à appeler, placez vous sous Word en mode "Enregistrement de macros" : Menu **"Outils" – "Macro" – "Nouvelle macro"**.

Faites les opérations désirées au clavier et souris.

Arrêtez l'enregistrement de macro, et allez voir le code généré par Word : Menu **"Outils" – "Macro" – "Macros" – "Modifier"**.

Exemple :

La macro qui active le graphique et modifie une propriété est la suivante :

```

ActiveSheet.ChartObjects("Graphique 1").Activate
ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowPercent,
LegendKey:=True _
, HasLeaderLines:=False
    
```

Note : Il y a 2 différences à noter entre la syntaxe VB d'Excel et la syntaxe LAUNCHER :

1) Quand on sélectionne un objet d'une collection, il faut ajouter ".Item" après le nom de la collection.

Pour sélectionner "Graphique 1" dans la collection des graphiques, la syntaxe VB :

```
ActiveSheet.ChartObjects("Graphique 1")
```

Devient

```
ActiveSheet.ChartObjects.Item("Graphique 1")
```

2) Les noms de paramètres (Nom:=valeur) est propre à la syntaxe VB.

Avec LAUNCHER, il faut lister les valeurs de chaque paramètre, dans l'ordre attendu, séparés par ';'.

L'instruction suivante

```
ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowPercent,  
LegendKey:=True
```

Devient

```
ActiveChart.ApplyDataLabels(xlDataLabelsShowPercent;True)
```

Sinon, optez pour l'écriture d'une macro sous Word, que vous appellerez par [XLEXEMACRO](#).

Exemple

L'exemple ci-dessous active la feuille 2 du classeur Excel

```
CHGVAR      VAR(&CMD) VALUE('XLMETHOD')  
CHGVAR      VAR(&PARM1) VALUE('Sheets.Item("Feuil2").Activate')  
CHGVAR      VAR(&PARM2) VALUE(' ')  
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +  
                        &PARM2 &RESULT))
```

Voir aussi

- [XLEXEMACRO](#)

Commande *XLMINIMIZE*

Permet de minimiser la fenêtre EXCEL courante.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('XLMINIMIZE')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLMAXIMIZE](#)

Commande *XLOPENFILE*

Ouvre un classeur EXCEL.

L'application Excel doit d'abord avoir été ouverte avec [EXCELOPEN](#).

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLOPENFILE')
CHGVAR          VAR(&PARM1) VALUE('Chemin du fichier')
CHGVAR          VAR(&PARM2) VALUE('
[TEXT=True/False]
[;ReadOnly=True/False]
[;Visible=True/False]
[;New=True/False]
[;AddToWorkbook=True/False]
[;Name="Nom de la feuille"]
[;PWD="password"]
')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Chemin et nom du fichier à ouvrir. Si seul le fichier est spécifié, il sera alors ouvert dans le répertoire par défaut des documents EXCEL. |
| Parm2 | <p>Parm2 permet de fixer des propriétés pour l'ouverture.</p> <p><u>Propriétés:</u></p> <p>Text : pour ouvrir le fichier en mode texte. Le fichier doit être du même format que celui créé par LAUNCHER Office (format CSV). L'option '*TEXT' est aussi accepté pour compatibilité avec les anciennes versions.</p> <p>ReadOnly : pour ouvrir le fichier en lecture seule. L'option '*ONLY' est aussi accepté pour compatibilité avec les anciennes versions.</p> <p>AddToWorkbook : pour ajouter le fichier lu au classeur courant, sous forme d'une nouvelle feuille. Si aucun classeur n'est ouvert, un nouveau est créé.</p> <p>Name : permet de donner un nom à la nouvelle feuille du classeur,</p> |

si l'option **AddToWorkbook** est aussi choisie.

New : crée un classeur vierge.

Visible : rend visible l'instance de Excel.

PWD : chaîne de caractères qui contient le mot de passe requis pour ouvrir un classeur protégé. Si cet argument est omis et que le classeur nécessite un mot de passe, le classeur est rendu visible, et l'utilisateur est invité à le saisir.

Remarque

En spécifiant l'option **Text**, LAUNCHER Office va rechercher s'il existe un fichier LFD, qui aurait été créé lors du transfert de fichier. Celui-ci reprend la description du fichier AS/400 pour la reproduire dans le classeur.

Exemples

```
CHGVAR      VAR(&CMD) VALUE('XLOPENFILE')
CHGVAR      VAR(&PARM1) VALUE('%LNCDIR%\Docs\fichier.xlsx')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Dans cet exemple, le fichier sera ouvert depuis le répertoire d'installation de Launcher.

```
CHGVAR      VAR(&CMD) VALUE('XLOPENFILE')
CHGVAR      VAR(&PARM1) VALUE(' ')
CHGVAR      VAR(&PARM2) VALUE('New;Visible')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Dans cet exemple, un nouveau classeur sera créé et rendu visible.

```
LNCCMD      CMD(EXCELOPEN)
LNCCMD      CMD(XLOPENFILE) PARM1('C:\A\base.xlsx')
```

Dans cet exemple, un classeur est ouvert après avoir créé une instance de l'application Excel.

```
LNCCMD      CMD(EXCELOPEN)
LNCCMD      CMD(XLOPENFILE) PARM1('%TEMP%\LNC002.TXT') +
PARM2('AddToWorkbook;Name="Customers"')
```

Dans l'exemple ci-dessus, le fichier "LNC002.TXT" est ajouté au classeur actuel sous le nom "Customers".

```
LNCCMD      CMD (EXCELOPEN)
LNCCMD CMD (XLOPENFILE) PARM1 ('C:\A\protected_workbook.xlsx')
PARM2 ('PWD="aura"')
```

Dans l'exemple ci-dessus, on ouvre un classeur protégé en spécifiant le mot de passe.

Voir aussi

- [XLCLOFILE](#)
- [XLSAVEAS](#)

Commande *XLPASTE*

Permet de copier le contenu du presse-papier vers la cellule active.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('XLPASTE')
CHGVAR          VAR (&PARM2) VALUE (' ')
CHGVAR          VAR (&PARM1) VALUE (' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2+
&RESULT) )
MONMSG          MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [XLCOPY](#)

Commande XLPRINT

Permet d'imprimer un classeur Excel.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLPRINT')
CHGVAR          VAR(&PARM1) VALUE('
[Printer="Nom_imprimante"]
[;Copies="Nombre_de_copies"]
[;From=Numéro]
[;To= Numéro]
[;Preview=True/False]
[;PrintToFile="Chemin_fichier"]
[;Collate=True/False]
[;Workbook=True/False]
[;Sheet=True/False]
[;Selection= True/False]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Printer est le nom de l'imprimante. Si Printer n'est pas spécifié, c'est l'imprimante par défaut qui est utilisée (voir Panneau de Configuration Windows).</p> <p>Copies est le nombre d'exemplaires à imprimer.</p> <p>From est le numéro de la première page à imprimer. From fait référence aux pages imprimées et non aux pages globales de la feuille ou du classeur.</p> <p>To est le numéro de la dernière page à imprimer. To fait référence aux pages imprimées et non aux pages globales de la feuille ou du classeur.</p> <p>Preview pour obtenir un aperçu avant impression.</p> <p>PrintToFile permet d'imprimer vers un fichier. Il faut spécifier le chemin et le nom du fichier.</p> <p>Collate permet d'avoir des copies assemblées. Organise les pages numérotées lorsque vous imprimez plusieurs copies d'un document.</p> |



Une copie complète du document est imprimée avant l'impression de la première page de la deuxième copie.

Workbook ou **Sheet** ou **Selection** pour sélectionner l'objet à imprimer.

Exemples

```
CHGVAR          VAR(&CMD) VALUE('XLPRINT')
CHGVAR          VAR(&PARM1) VALUE('Printer="Brother HL-1270N series";
                                From=2;"To=8;Sheet')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
```

Les pages 2 à 8 de la feuille courante sont imprimées sur l'imprimante "Brother HL-1270N series".

```
LNCCMD CMD(XLPRINT) PARM1('printer="LBP611C";Workbook')
```

Le classeur entier est imprimé sur l'imprimante "LBP611C".

Voir aussi

- [WPRINT](#)

Commande *XLPROTECT*

Protège la feuille courante contre les modifications à l'aide d'un mot de passe.

Toutes les cellules qui seront verrouillées (format de cellule) ne seront pas modifiables. Une tentative d'écriture de la part de l'utilisateur provoque l'affichage d'une boîte l'informant qu'il ne peut pas accéder au contenu de la cellule, ni à ses propriétés.

Les cellules préalablement déverrouillées par l'utilisateur seront modifiables.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('XLPROTECT')
CHGVAR          VAR(&PARAM1) VALUE('MotDePasse')
CHGVAR          VAR(&PARAM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
                               &PARAM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | Mot de passe de protection de la feuille courante. A rappeler pour ôter la protection. |

Exemple

```
CHGVAR          VAR(&CMD) VALUE('XLPROTECT')
CHGVAR          VAR(&PARAM1) VALUE('Aura')
CHGVAR          VAR(&PARAM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARAM1 +
                               &PARAM2 &RESULT)
```

Voir aussi

- [XLDPROTECT](#)

Commande *XLREPLACE*

Recherche et remplacement de texte dans la classeur Excel courant.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLREPLACE')
CHGVAR          VAR(&PARM1) VALUE('
                ;Find="Texte recherché"
                ;ReplaceWith="Texte de remplacement"
                [;MatchCase]
                ')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Find est le texte recherché.</p> <p>ReplaceWith est le texte de remplacement.</p> <p>MatchCase indique que la recherche sera sensible à la casse (distinction majuscules et minuscules).</p> |

Exemple

```
LNCCMD CMD(XLREPLACE) PARM1('Find="Aura";ReplaceWith="Equipements"')
```

Recherche les cellules contenant "Aura"/"aura" et remplacement de "Aura"/"aura" par "Equipements".

```
LNCCMD CMD(XLREPLACE) +
PARM1('Find="Aura";ReplaceWith="Equipements";Matchcase')
```

Recherche (sensible à la casse) les cellules contenant "Aura" et remplacement de "Aura" par "Equipements".

Voir aussi

- [XLFIND](#)

Commande XLRESIZE

Cette commande insère ou supprime des lignes et des colonnes, afin de donner une nouvelle taille à une zone d'une feuille Excel.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLRESIZE')
CHGVAR          VAR(&PARM1) VALUE('
[Ref="Référence de cellules"]
[;RowCount=nombre]
[;ColCount=nombre]
[;Rowcnt=nombre]
[;Colcnt=nombre]
[;EntireRow=True/False]
[;EntireColumn=True/False]
[;Clear=True/False]
[;Fill=True/False]
[;Name="Nouveau nom de zone"]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 + &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Ref : La référence de la zone à redimensionner.</p> <ol style="list-style-type: none"> Soit on spécifie le nom de la zone. Soit on spécifie la référence de la cellule en haut à gauche de la zone à redimensionner. <p>La référence de la cellule en haut à gauche peut être donnée sous la forme :</p> <ul style="list-style-type: none"> - Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3. - Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B). - Nom symbolique. - "." pour désigner la cellule active. <p>Dans ce cas-là on peut utiliser Rowcnt et Colcnt pour spécifier la référence de la cellule en bas à droite de la zone à redimensionner.</p> <p>RowCnt , ColCnt : on utilise Rowcnt et Colcnt pour spécifier la référence de la cellule en bas à droite de la zone à redimensionner, si on a spécifié la référence de la cellule en haut à gauche de la zone à redimensionner avec le paramètre Ref. Dans ce cas-là, Rowcnt et Colcnt représentent respectivement le nombre de lignes et de colonnes sélectionnées, à partir de la référence données par Ref.</p> <p>RowCount : Nombre de lignes de la zone redimensionnée.</p> |

Si **RowCount** est absent, le nombre de lignes de la zone n'est pas changé.

ColCount : Nombre de colonnes de la zone redimensionnée.
Si **ColCount** est absent, le nombre de colonnes de la zone n'est pas changé.

EntireRow : *True* pour insérer ou supprimer des lignes entières, ou *False* (défaut) pour décaler les cellules vers le haut ou bas sur la largeur de la sélection.

EntireColumn : *True* pour insérer ou supprimer des colonnes entières, ou *False* (défaut) pour décaler les cellules vers la gauche ou la droite sur la hauteur de la sélection.

Fill : *True* pour copier la sélection sur l'ensemble des nouvelles cellules insérées.
False (défaut) pour laisser les nouvelles cellules vides.
Cette option permet de copier les formules de calculs dans les nouvelles lignes ou colonnes insérées.

Clear : *True* efface toute les cellules ne contenant pas de formules, dans la nouvelle zone.

Name permet d'affecter un nom à la zone redimensionnées. Si le **Ref** fait référence à une zone nommée, le même nom peut être donné à **Name**.

Exemples

L'exemple suivant redimensionne la zone nommée "Commandes", à 10 lignes.
Le nombre de colonnes de la zone ne change pas.
Des lignes entières sont supprimées ou ajoutées.

```
CHGVAR      VAR(&CMD) VALUE('XLRESIZE')
CHGVAR      VAR(&PARM1)
             VALUE('Ref="Commandes";RowCount=10;EntireRow=True')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                             &PARM2 &RESULT)
```

La zone nommée "Commandes" est redimensionnée à 10 lignes.
Les formules sont copiées vers les nouvelles lignes.
Les cellules ne contenant pas de formules sont effacées.
Les insertions ou suppressions ne se font que sur la largeur de la zone.

```
CHGVAR      VAR(&CMD) VALUE('XLRESIZE')
CHGVAR      VAR(&PARM1) +
             VALUE('Ref="Commandes";RowCount=10; +
```

```
Fill=True;Clear=True')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

La zone suivante :

| | A | B | C |
|---|---|---|---|
| 1 | 1 | 2 | 3 |

est redimensionnée avec 10 lignes, et nommée "Data":

```
LNCCMD CMD(XLRESIZE) +
PARM1('Ref="$A$1";RowCount=10;name="Data";Fill=false;Rowcnt=3;Colcnt
=3')
```

Voir aussi

- [XLCELLS](#)

Commande XLSAVEAS

Sauvegarde le classeur EXCEL courant.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLSAVEAS')
CHGVAR          VAR(&PARM1) VALUE('[fichier]')
CHGVAR          VAR(&PARM2) VALUE('
                [Format=constante]
                [;Replace=True/False]
                [;Backup=True/False]
                ')
CHGVAR          VAR(&PARM2) VALUE('Options')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Chemin et nom de sauvegarde du classeur courant. Si ce paramètre est vide, le document sera sauvé avec le nom sous lequel il a été ouvert.</p> |
| Parm2 | <p>Format=format de sauvegarde du document. Le format doit être cohérent avec l'extension du fichier sauvegardé. Vous pouvez utiliser la syntaxe Excel:</p> <p>xlAddIn , xlCSV, xlCSVMac, xlCSVMSDOS, xlCSVWindows, xlDBF2, xlDBF3, xlDBF4, xlDIF, xlExcel2, xlExcel3, xlExcel4, xlExcel4Workbook, xlExcel5, xlExcel7, xlExcel9795, xlHTML, xlIntlAddIn, xlIntlMacro...</p> <p>ou bien les mots clés suivants :</p> <ul style="list-style-type: none"> - XLS - XLSX - XLSM <p><u>Remarque</u> : si vous ne mettez pas le paramètre Format, alors le format par défaut de MS Excel est utilisé. Exemple : XLSX pour Excel 2016.</p> <p>Replace=True/False. False par défaut. Remplace le fichier s'il existe si Replace=True.</p> <p>Backup=True/False. False par défaut. True pour créer un fichier de backup.</p> |

Exemples

1)

```
LNCCMD CMD (XLSAVEAS) PARM1 ('c:\temp\res.xlsx') PARM2 ('format=XLSX')
```

2)

```
LNCCMD CMD (XLSAVEAS) PARM1 ('c:\temp\res.xlsm') PARM2 ('format=XLSM')
```

3)

```
CHGVAR          VAR(&CMD) VALUE('XLSAVEAS')
CHGVAR          VAR(&PARM1) VALUE('c:\chemin\sauvegarde.xlsx')
CHGVAR          VAR(&PARM2) VALUE('Replace')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                              &PARM2 &RESULT)
```

Voir aussi

- [XLADDSHEET](#)
- [XLDELSHEET](#)
- [XLGOTOSH](#)

Commande *XLSETLINE*

Affecte les valeurs à plusieurs cellules sur une même ligne, à partir de la cellule active.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('XLSETLINE')
CHGVAR          VAR(&PARM1) VALUE('Valeurs')
CHGVAR          VAR(&PARM2) VALUE('Propriétés')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Chaîne de caractères contenant les valeurs à affecter aux cellules.</p> <p>La chaîne de caractères peut contenir le mot symbolique suivant : %SEP% : Pour passer à la cellule suivante. Si on était positionné sur la dernière colonne du tableau, on passe au début de la ligne suivante.</p> |
| Parm2 | <p>Chaîne de caractères pouvant contenir les formats des cellules.</p> <p>Les formats de chaque cellule sont séparés par le mot %SEP%. On peut appliquer plusieurs formats sur une cellule. Ils sont alors séparés par un point virgule.</p> <p><u>Les formats peuvent être les suivants :</u></p> <p>NUMFMT(d [DECPOINT=p] [GRPOINT=g])</p> <p>Où :</p> <ul style="list-style-type: none"> -d est le nombre de décimales, -p représente le caractère à utiliser pour le point décimal. (Caractère point par défaut), -g est le caractère à utiliser pour séparer les groupes de 3 chiffres. <p>PROP fixe la valeur d'une propriété pour le contenu de la cellule correspondante.</p> <p>Exemple :</p> <p><i>PROP(Font.bold)=True; PROP(Font.italic)=True</i></p> |

Exemples

L'exemple ci-dessous envoie les valeurs pour 4 cellules, et fixe la couleur Bleue à la deuxième cellule, dont la valeur est envoyée en format numérique, avec 2 décimales.

```
CHGVAR      VAR (&CMD) VALUE ('XLSETLINE')
CHGVAR      VAR (&PARM1) VALUE (&NAME *CAT '%SEP%' *CAT +
&AMOUNT *CAT '%SEP%' *CAT &ADDRESS *CAT '%SEP%' +
*CAT &CITY)
CHGVAR      VAR (&PARM2) VALUE ('%SEP%NUMFMT (2);PROP (Font.Color)= +
INT (255) ')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

L'exemple ci-dessous crée un nouveau classeur et insert 2 lignes à partir de la cellule B2 :

```
LNCCMD      CMD (EXCELOPEN) PARM1 ('visible;new')
LNCCMD      CMD (XLCELLS) PARM1 ('Ref="B$2";select=true')
LNCCMD      CMD (XLSETLINE) +
PARM1 ('Aura%SEP%Equipements%SEP%2019') +
PARM2 ('%SEP%PROP (Font.Bold)=true;PROP (Font.+
Color)=INT (255) %SEP%PROP (Font.Color)=vbBlue')
LNCCMD      CMD (XLINSERT) PARM1 ('After=true')
LNCCMD      CMD (XLSETLINE) +
PARM1 ('Power8%SEP%IBM%SEP%2019') +
PARM2 ('%SEP%PROP (Font.Bold)=TRUE;PROP (Font.+
Color)=INT (255) %SEP%PROP (Font.Color)=vbBlue')
```

| | A | B | C | D | E |
|---|---|--------|----------|------|---|
| 1 | | | | | |
| 2 | | Aura | Equipeme | 2019 | |
| 3 | | Power8 | IBM | 2019 | |
| 4 | | | | | |
| 5 | | | | | |

Note

Pour utiliser la propriété **Font.Color**, il faut utiliser les codes couleur Windows. Soit les constantes:

| Constant | Value | Description |
|------------------|----------|-------------|
| vbBlack | &H0 | Black |
| vbRed | &HFF | Red |
| vbGreen | &HFF00 | Green |
| vbYellow | &HFFFF | Yellow |
| vbBlue | &HFF0000 | Blue |
| vbMagenta | &HFF00FF | Magenta |
| vbCyan | &HFFFF00 | Cyan |
| vbWhite | &HFFFFFF | White |

Soit les valeurs:

| | |
|------------------|-----------------|
| vbBlack | 0 |
| vbBlue | 16711680 |
| vbCyan | 16776960 |
| vbGreen | 65280 |
| vbMagenta | 16711935 |
| vbRed | 255 |
| vbWhite | 16777215 |
| vbYellow | 65535 |

PROP(Font.Color)=vbMagenta
PROP(Font.Color)=INT(255) //pour rouge

On peut aussi utiliser les valeurs BGR avec la fonction RGB : RGB(blue,green,red).

PROP(Font.Color)=RGB(255,0,0) //bleu
 PROP(Font.Color)=RGB(0,255,0) // vert
 PROP(Font.Color)=RGB(0,0,255) // rouge

Exemple :

```
LNCCMD CMD (XLGOTOCELL) PARM1 (' $D$5')
```

```
LNCCMD CMD (XLSETLINE) PARM1 ('A%SEP%B%SEP%C') +
  PARM2 ('PROP (Font.Color)=RGB (0, 255, 0) ; PROP (Font.Bold)=True+
  %SEP%PROP (Font.Color)=RGB (255, 0, 0) %SEP%PROP (Font.Color)=RGB (0, 0, 255) '
  )
```

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | A | B | C |
| 6 | | | | | | |
| 7 | | | | | | |



Commande XLSETPROP

Fixe la valeur d'une propriété de divers objets (classeur, feuille, cellule ou sélection) sur une instance d'Excel.

Syntaxe

```
CHGVAR          VAR(&CMD) VALUE('XLSETPROP')
CHGVAR          VAR(&PARM1) VALUE('Propriété')
CHGVAR          VAR(&PARM2) VALUE('Valeur')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | <p>Le chemin complet vers la propriété à modifier. Exemple : ActiveCell.Font.Color</p> <p><u>Quelques exemples :</u> ActiveCell.Font.Color : <i>Change la couleur.</i> ActiveCell.Font.Bold : <i>Permet de mettre en gras/non gras la sélection.</i></p> |
| Parm2 | <p>Valeur à affecter à la propriété.</p> <p><u>Exemples :</u></p> <p>TRUE : Valeur Vrai. FALSE : Valeur Faux. INT(chaine) : Quand Excel attend une valeur entière.</p> |

Note

Le **chemin vers la propriété** respecte la syntaxe utilisée en Visual Basic.

On peut utiliser les valeurs constantes de Excel, telles que xlBottom, xlVAlignCenter.

Quand le chemin vers la propriété inclut un élément d'une collection (tel que `ActiveCell.Borders...`, qui a 4 bordures), il faut préciser l'élément concerné (Item).

Par exemple, dans le cas des bordures, en Visual Basic, la propriété est exprimée par :

```
ActiveCell.Borders(xlBottom).LineStyle = xlDashDot
```

En Visual Basic, Item est une propriété par défaut qui ne figure pas dans la syntaxe.

Exemple

L'exemple ci-dessous fixe la couleur de police de la cellule active en rouge :

```
CHGVAR      VAR(&CMD) VALUE('XLSETPROP')
CHGVAR      VAR(&PARM1) VALUE('ActiveCell.Font.Color')
CHGVAR      VAR(&PARM2) VALUE('INT(255)')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Commande *XLSETTEXT*

Affecte du texte à une cellule de la feuille Excel courante.

Le format de la cellule est défini comme Texte.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('XLSETTEXT')
CHGVAR          VAR (&PARM1) VALUE ('Coordonnées')
CHGVAR          VAR (&PARM2) VALUE ('Texte')
CALL           PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Coordonnées de la cellule à modifier.</p> <p>Les coordonnées de la cellule peuvent être données sous la forme :</p> <ul style="list-style-type: none"> - Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3. - Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B). - Nom symbolique donné à une cellule ou un groupe de cellules. <p>Si Parm1 n'est pas renseigné, c'est la cellule active qui sera modifiée.</p> |
| Parm2 | <p>Texte à affecter.</p> |

Exemples

```
CHGVAR          VAR (&CMD) VALUE ('XLSETTEXT')
CHGVAR          VAR (&PARM1) VALUE ('$A$1')
CHGVAR          VAR (&PARM2) VALUE ('Dépenses')
CALL           PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

OU

```
LNCCMD CMD(XLSETTEXT) PARM1('C7') PARM2('Aura')
```

OU

```
LNCCMD CMD(XLSETTEXT) PARM1('Name') PARM2('Aura')
```

ou (modification cellule active)

```
LNCMD CMD (XLSETTEXT) PARM2 ('Aura')
```

Voir aussi

- [XLCELLNAME](#)
- [XLGOTOCELL](#)
- [XLSETVALUE](#)
- [XLGETVALUE](#)

Commande *XLSETVALUE*

Affecte une valeur à une cellule de la feuille Excel courante.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('XLSETVALUE')
CHGVAR      VAR (&PARM1) VALUE ('Coordonnées')
CHGVAR      VAR (&PARM2) VALUE ('Valeur')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|---|
| Parm1 | Coordonnées de la cellule à modifier. Les coordonnées de la cellule peuvent être données sous la forme : <ul style="list-style-type: none">- Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3.- Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B).- Nom symbolique donné à une cellule ou un groupe de cellules. Si les coordonnées de la cellule ne sont pas renseignées, c'est la cellule active qui sera modifiée. |
| Parm2 | Valeur à affecter. |

Exemple

```
CHGVAR      VAR (&CMD) VALUE ('XLSETVALUE')
CHGVAR      VAR (&PARM1) VALUE ('Total')
CHGVAR      VAR (&PARM3) VALUE ('1000')
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XLCELLNAME](#)
- [XLGOTOCELL](#)
- [XLSETTEXT](#)
- [XLGETVALUE](#)

- [XLCELLS](#)

Commande *XLSHNAME*

Change le nom de la feuille courante Excel.

Syntaxe

```
CHGVAR      VAR (&CMD) VALUE ('XLSHNAME ')  
CHGVAR      VAR (&PARM1) VALUE ('Nom ')  
CHGVAR      VAR (&PARM2) VALUE (' ')  
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +  
                        &PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|-------------------------------------|
| Parm1 | Nouveau nom de la feuille courante. |

Exemple

```
CHGVAR      VAR (&CMD) VALUE ('XLSHNAME ')  
CHGVAR      VAR (&PARM1) VALUE ('Data ')  
CHGVAR      VAR (&PARM2) VALUE (' ')  
CALL        PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +  
                        &PARM2 &RESULT)
```

ou

```
LNCCMD CMD (XLSHNAME) PARM1 ('data')
```

Voir aussi

- [XLADDSHEET](#)
- [XLDELSHEET](#)
- [XLGOTOSH](#)

Commande XLSORT

Permet de trier les cellules d'une feuille selon une ou plusieurs colonnes.

Syntaxe

```

CHGVAR      VAR(&CMD) VALUE('XLSUBTOTAL')
CHGVAR      VAR(&PARM1) VALUE('
[;Sheet="Nom de la feuille"]
[;Key1=cellule]
[;Order1=xlDescending/xlAscending]
[;Key2=cellule]
[;Order2=xlDescending/xlAscending]
[;Key3=cellule]
[;Order3=xlDescending/xlAscending]
[;Header=xlYes/xlNo]
')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>Sheet : Indique le nom de la feuille à traiter. Si le mot clé Sheet est absent, c'est la feuille active qui est prise en compte. Toutes les cellules de la feuille seront triées.</p> <p>Key1, Key2, Key3 : Coordonnées de la 1^{ère} cellule de la colonne à partir de laquelle on veut effectuer le tri. Les coordonnées de la cellule peuvent être données sous la forme : - Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3. - Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B). - Nom symbolique.</p> <p>Key1 sera l'ordre de tri prioritaire, suivi par Key2 et enfin par Key3.</p> <p>Order1, Order2, Order3 : Ordre du tri. Correspond respectivement à Key1, Key2 et Key3. Les valeurs peuvent être : xlDescending : ordre décroissant de valeurs. xlAscending : ordre croissant de valeurs. Valeur par défaut : xlAscending_</p> |

Header : spécifie si la feuille contient une ligne d'entêtes des colonnes, afin qu'elle ne fasse pas partie du tri.

xlYes : la feuille contient une ligne d'entêtes des colonnes, et on veut exclure cette ligne du tri.

xlNo : la feuille ne contient pas de lignes d'entêtes, ou bien on veut que la ligne d'entêtes fasse partie du tri. Valeur par défaut.

Exemple

On veut trier les cellules de la feuille active selon les valeurs de la colonne H (valeurs décroissantes), et ensuite, le résultat de ce premier tri sera ordonné selon les valeurs de la colonne A (valeurs croissantes).

La première ligne d'entêtes des colonnes ne sera pas prise en compte pour les tris.

```
LNCCMD      CMD (EXCELOPEN) +  
            PARM1 ('C:\A\sp_cust_ref.xlsx') +  
            PARM2 ('visible')
```

```
LNCCMD      CMD (XLSORT) +  
            PARM1 ('Key2="$A$2";Order2=xlAscending;Key1=+  
"$H$2";Order1=xlDescending;Header=xlYes')
```

Commande XLSUBTOTAL

Permet d'appliquer une fonction de totalisation par groupe de lignes, sur des colonnes dans une sélection.

Syntaxe

```

CHGVAR          VAR(&CMD) VALUE('XLSUBTOTAL')
CHGVAR          VAR(&PARM1) VALUE('
TotalList="liste"
[;Ref="Référence de cellules sélectionnées"]
[;Sheet="Nom de la feuille"]
[;RefTo="Référence de fin de zone sélectionnée"]
[;GroupBy=Rang de colonne]
[;Function=Code de fonction]
[;Replace=True/False]
[;PageBreaks=True/False]
[;SummaryBelowData=xlSummaryAbove/xlSummaryBelow]
')
CHGVAR          VAR(&PARM2) VALUE(' ')
CALL            PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
    
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | <p>TotalList : Liste des rangs des colonnes sur lesquelles la fonction doit être appliquée (paramètre Function). Les colonnes sont numérotées de 1 à n. Les numéros sont séparés par le caractère "point virgule", dans une liste entre guillemets. Obligatoire.</p> <p>Ref : La référence de la zone à traiter.</p> <p>1) Soit on spécifie le nom de la zone. 2) Soit on spécifie la référence de la cellule en haut à gauche de la zone à redimensionner.</p> <p>La référence de la cellule en haut à gauche peut être donnée sous la forme :</p> <ul style="list-style-type: none"> - Syntaxe \$B\$3. Exemple : \$B\$3 désigne la colonne B, ligne 3. - Syntaxe CL. C pour désigner la colonne et L pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B). - Nom symbolique. <p>Dans ce cas-là on utilise RefTo pour spécifier la référence de la cellule en bas à droite de la zone à traiter.</p> <p>Si Ref est absent, toutes les cellules de la feuille seront traitées.</p> <p>Sheet : Indique le nom de la feuille à traiter.</p> |

Si **Sheet** est absent, c'est la feuille active qui sera traitée.

RefTo : Référence à la cellule en bas et à droite de la zone sélectionnée. Cette référence est donnée sous la forme :

- Syntaxe **\$B\$3**. Exemple : \$B\$3 désigne la colonne B, ligne 3.
- Syntaxe **CL. C** pour désigner la colonne et **L** pour désigner la ligne. Exemple : B3 désigne la ligne 3, et la colonne 2 (B).
- Nom symbolique.

GroupBy : Cette option indique le rang de la colonne à utiliser pour regrouper les totalisations. Les colonnes sont numérotées de 1 à n. Valeur par défaut = **1**.

Function : Désigne la fonction à appliquer.

Les valeurs possibles sont :

xIAverage, xICount, xICountNums, xIMax, xIMin, xIProduct, xIStDev, xIStDevP, xISum, xIUnknown, xIVar, xIVarP.

Consultez [la documentation de Microsoft Excel](#) pour plus de détail sur ces fonctions.

Valeur par défaut : **xISum**.

Replace : False pour ne pas remplacer les actuels sous-totaux. Par défaut, les sous-totaux sont remplacés (True).

PageBreaks : Pour ajouter un saut de page après chaque groupe. Valeur par défaut : False.

SummaryBelowData : Permet de choisir la position des sous-totaux.

Les valeurs possibles sont :

xISummaryAbove pour placer les sous-totaux avant chaque groupe de données.

xISummaryBelow (Défaut) pour placer les sous-totaux après les données du groupe.

Exemple

Dans l'exemple suivant, des sous-totaux sont créés pour la colonne 12, avec des regroupements opérés en fonction des colonnes 1 et 6.

L'ensemble des cellules de la feuille "data" sont prises en compte.

PGM



```
LNCOPEN  
  
LNCCMD CMD (EXCELOPEN)  
  
LNCCMD CMD (XLOPENFILE) PARM1 ('C:\A\DataRef.XLSX')  
  
LNCCMD CMD (XLGOTOSH) PARM1 ('data')  
  
LNCCMD CMD (XLSUBTOTAL) +  
PARM1 ('sheet="data";GroupBy=1;Function=xlSu+  
m;TotalList="12"')  
  
LNCCMD CMD (XLSUBTOTAL) +  
PARM1 ('sheet="data";GroupBy=6;Function=xlSu+  
m;TotalList="12";REPLACE=FALSE')  
  
LNCCMD CMD (XLSAVEAS) +  
PARM1 ('C:\A\RES.XLSX;replace=true')  
  
LNCCMD CMD (EXCELCLOSE)  
  
LNCCLOSE  
  
ENDPGM
```

Pour n'appliquer les sous-totaux que sur une sélection de cellules, on peut utiliser la commande suivante :

```
LNCCMD      CMD (XLSUBTOTAL) +  
            PARM1 ('Ref="A1";RefTo="N14";sheet="data";Gr+  
                oupBy=1;Function=xlSum;TotalList="12"')
```

Commande *XLUNDERLN*

Passer le contenu de la cellule active en caractères soulignés.

Syntaxe

```
CHGVAR          VAR (&CMD) VALUE ('XLUNDERLN')
CHGVAR          VAR (&PARM1) VALUE (' ')
CHGVAR          VAR (&PARM2) VALUE (' ')
CALL            PGM (LNCCMD) PARM (&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
```

Voir aussi

- [XLBOLD](#)
- [XLITALIC](#)

Commande *XLWAIT*

Attend un événement EXCEL.

Le programme AS/400 attend la fermeture d'Excel (**EXCELCLOSE**), ou une action sur le menu personnalisé (**XMLMENU**), pour continuer son exécution.

Syntaxe

```
CHGVAR      VAR(&CMD) VALUE('XLWAIT')
CHGVAR      VAR(&PARM1) VALUE('[BackPrint=Valeur]')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Paramètres

| Paramètres | |
|------------|--|
| Parm1 | La propriété BackPrint permet d'attendre qu'une impression en arrière plan se termine. La valeur est exprimée en secondes. 0 = infini. |

Exemple

```
CHGVAR      VAR(&CMD) VALUE('XLWAIT')
CHGVAR      VAR(&PARM1) VALUE('BackPrint=10')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
```

Voir aussi

- [XMLMENU](#)
- [EXCELCLOSE](#)

Les commandes CL

Généralités

Il est possible de développer des applications qui n'utilisent que les commandes CL de haut niveau, sans utiliser les programmes API.

La conversation entre le programme AS/400 et le PC est gérée par les commandes CL, au travers des paramètres **EXESRV** et **ENDOPT** des commandes CL.

Le paramètre **EXESRV** peut prendre les valeurs :

- ***DEV** pour désigner le PC utilisé pour l'émulation de terminal.
- ***CURRENT** pour désigner la conversation déjà ouverte. Dans ce cas, une conversation doit être ouverte au préalable, par le programme LNCOPEN, ou par une autre commande CL qui a maintenu sa conversation ouverte (**ENDOPT=*NONE**).
- **"Adresse IP/Nom DNS"** pour désigner un PC par son adresse IP ou son nom.

Le paramètre **ENDOPT** peut prendre les valeurs :

- ***NONE** pour tout garder ouvert : le document, l'application et la conversation.
- ***DOC** pour ne fermer que le document.
- ***APP** pour fermer le document et l'application.
- ***ALL** pour tout fermer, y compris la conversation.

Valeurs par défaut (WRKLNCDFE)

Grâce à la commande WRKLNCDFE, vous avez accès à toutes les valeurs par défaut utilisées par le client Launcher/400, quand la valeur *DFE est utilisée dans une commande CL:

- LNCDFR** Dossier utilisé par WORD par défaut
- LNCDFL** Chemin d'accès aux documents
- LNCDFN** Chemin d'accès + nom du document généré pour les sorties PRN
- LNCDFP** Imprimante utilisée par défaut par Launcher/400
- LNCDFR** Valeur par défaut pour le paramètre LNCDFR
- LNCDFV** Chemin d'accès pour sauvegarder les documents (LNCDFDOC)
- LNCDFE** Chemin d'accès utilisé pour ctrl invite LNCDFDOC
- LNCDFV** Serveur utilisé par défaut par la commande LNCDFDOC
- LNCDFR** Chemin d'accès au dossier des fichiers transférés
- LNCDFP** Nom et chemin d'accès au fichier de transfert temporaire
- LNCDFH** Chemin d'accès aux documents ACCESS
- LNCDFV** Serveur utilisé par défaut par la commande CPYTOMDB
- LNCDFG** Chemin d'accès et nom du fichier Log pour SMTP
- LNCDFL** Adresse email de retour
- LNCDFM** Serveur SMTP utilisé par la commande LNCDFNDMAIL
- LNCDFN** Adresse email utilisé par défaut pour les envois messagerie
- LNCDFV** Serveur Launcher utilisé par la commande LNCDFNDMAIL
- LNCDFH** Chemin d'accès aux documents utilisé par la commande CPYTOXLS
- LNCDFV** Chemin d'accès pour sauvegarder les tableurs
- LNCDFV** Serveur utilisé par défaut pour CPYTOXLS

CPYTOMDB - Commande CL

La commande **CPYTOMDB** permet de transférer des données d'un fichier ou d'une requête de l'AS/400 vers une table dans une base de données ACCESS.

Paramètres

| | | |
|------------------------------|-----------------|-----------------|
| Sélection auto base ACCESS | SELMDB | F |
| Base ACCES de destination | TOMDB | |
| Chemin d'accès à la base | MDBPTH | *DFT |
| Source de la copie | CPYSRC | *FILE *SQL |
| Fichier d'origine | FROMFILE | |
| Bibliothèque | | *LIBL |
| Membre d'origine | FROMMBR | *FIRST |
| Requête SQL | FROMSQL | 'SELECT * FROM' |
| Analyse | FROMQRY | |
| Bibliothèque | | *LIBL |
| Table ACCESS de destination | TOTBL | |
| Remplacement ou ajout enreg. | TBLOPT | *ADD |
| Afficher le document | SHOWDOC | *YES |
| Transfert unicode | | *YES |
| Nom du serveur LAUNCHER | EXESRV | *DFT |
| Option de fin | ENDOPT | *NONE |

Détails

La commande **CPYTOMDB** permet de lancer l'application **ACCESS** sur un P.C. pour effectuer différentes copies dans les tables de la base de données active.

Les données peuvent être ajoutées aux enregistrements déjà contenus dans une table ou remplacer le contenu de la table.

La technique de la copie ***MAP *DROP** de CPYF est utilisée à chaque fois.

Sélection auto base ACCESS (SELMDB)

Ce paramètre permet de sélectionner le document manuellement (saisie obligatoire) ou automatiquement (sélection du document à partir d'une boîte de dialogue Windows).

Ce paramètre est obligatoire.

Les valeurs possibles sont :

F

La valeur **F** (comme FREE) permet à l'utilisateur d'obtenir l'invite pour saisir le nom du document, son dossier ainsi que son chemin d'accès.

S

La valeur **S** (comme SELECT) permet à l'utilisateur de sélectionner automatiquement le nom du document par l'intermédiaire d'une boîte de dialogue Windows renvoyée à

l'écran.

Il est donc possible dans ce cas de parcourir les différents dossiers de son disque local (ou disques réseaux) pour y retrouver la base de données ACCESS.

Remarque : L'utilisation du choix **S** n'est valable qu'à partir de la ligne de commandes AS/400. Lors de la création d'un programme CL, ce choix ne fonctionne pas et **F** est obligatoire.

Base ACCESS de destination (TOMDB)

Indique le nom de la base ACCESS à utiliser ou à créer.

La création de la base de données ACCESS est toujours effectuée si la base de données saisie au paramètre **TOMDB** n'existe pas.

Remarque : L'extension **.mdb** ne doit pas être spécifiée

Les valeurs possibles sont :

nom-base-de-données

Indiquez le nom de la base de données à utiliser ou à créer.

***CURRENT**

La base de donnée ouverte lors de la dernière session LAUNCHER Office est de nouveau utilisée. Cette option est valable lorsque l'utilisateur a besoin d'effectuer plusieurs copies de données dans différentes tables d'une même base. La valeur ***CURRENT** n'est valable que si la dernière connexion LAUNCHER Office n'a pas été fermée lors de la dernière commande CPYTOMDB.

Chemin d'accès à la base (MDBPTH)

Indique le nom du chemin d'accès à utiliser pour la base de données.

Les valeurs possibles sont les suivantes :

***DFT**

La valeur spéciale ***DFT** indique que le chemin d'accès utilisé pour l'accès à la base ACCESS est le chemin d'accès référencé dans le fichier des options par défaut LAUNCHER400. Pour visualiser ou modifier la valeur du chemin d'accès utilisé par défaut par la commande CPYTOMDB, sélectionnez dans la liste le mot clé **MDBPTH** après appel du programme **WRKLNCDFT**.

***NONE**

Indique que le chemin d'accès (root et dossier) est déjà indiqué au paramètre **TOMDB**.

nom-chemin-d'accès

Indiquez le nom complet du chemin d'accès utilisé pour accéder à la base ACCESS en respectant la syntaxe suivante : **C:\Program Files\LAUNCHER400\Mes Bases**

Remarque : La syntaxe **C:\Mes Documents\LAUNCHER400** est correcte alors que la syntaxe **C:\Mes Documents\LAUNCHER400** est incorrecte.

Source de la copie (CPYSRC)

Indique l'origine de la source à copier.

Les valeurs possibles sont les suivantes :

***FILE**

L'origine de la copie est un fichier base de données.

***QRY**



L'origine de la copie est une analyse query (objet de type *QRYDFN).

***SQL**

L'origine de la copie est une instruction SQL **SELECT**

Fichier d'origine (FROMFILE)

Indique le nom et la bibliothèque du fichier base de données à copier. Ce paramètre est utilisé lorsque le paramètre **CPYSRC** est défini sur ***FILE**.

Le fichier base de données peut être physique ou logique.

Pour la bibliothèque, les valeurs possibles sont les suivantes :

***LIBL**

La recherche est effectuée dans toutes les bibliothèques utilisateur et système de la liste des bibliothèques du travail jusqu'à ce que la première occurrence soit trouvée.

***CURLIB**

Le fichier base de données est recherché dans la bibliothèque en cours du travail, si celle-ci n'est pas précisée.

Membre d'origine (FROMMBR)

Indique le nom du membre fichier base de données, à utiliser pour la copie. Ce paramètre est utilisé lorsque le paramètre **CPYSRC** est défini sur ***FILE**.

Les valeurs possibles sont les suivantes :

***FIRST**

Le premier membre du fichier base de données est copié.

nom-membre

Indique le nom du membre du fichier base de données d'origine.

Requête SQL (FROMSQL)

Saisir la requête SQL pour extraire les données à copier.

This parameter is used when parameter **CPYSRC** is set to ***SQL**.

Analyse (FROMQRY)

Spécifie le nom de la requête à utiliser pour obtenir les données.

Ce paramètre est utilisé lorsque le paramètre **CPYSRC** est défini sur ***QRY**.

Nom de la requête

Entrez le nom de la requête.

Les valeurs de bibliothèque possible sont:

*** LIBL**

La liste de bibliothèques est explorée pour trouver la bibliothèque où se trouve le fichier de requête.

Nom de la bibliothèque

Entrez le nom de la bibliothèque contenant le fichier de requête.

Table ACCESS de destination (TOTBL)

Indique le nom de la table ACCESS utilisée.

Si la table n'existe pas, cette dernière est automatiquement créée.

Remplacement ou ajout enregistrements. (TBLOPT)



Indique comment les enregistrements sont introduits dans la table ACCESS.

Les valeurs possibles sont :

***ADD**

L'ajout des données est effectué sur la table ACCESS.

***REPLACE**

Le remplacement des données de la table ACCESS est effectué.

Afficher le document (SHOWDOC)

Indique si le document est affiché ou non à l'écran.

Les valeurs possibles sont :

***NO**

Le document résolu n'est pas montré à l'écran.

***YES**

Le document résolu sera rendu visible à l'écran.

Transfert Unicode

*YES (par défaut) ou *NO.

Nom du serveur LAUNCHER (EXESRV)

Spécifie le nom du serveur sur lequel ACCESS doit être exécuté. Ce nom peut correspondre à un autre poste de travail. Le nom du Host ou l'adresse IP sont des choix valides.

Les valeurs possibles sont :

***DFT**

Le traitement est exécuté sur le serveur référencé dans le fichier **LNCDFTP** réservé à la définition des options LAUNCHER Office (**mot clé MDBSRV**).

***DEV**

Le traitement est exécuté sur le poste du demandeur.

***CURRENT**

Dans le cas où la dernière connexion n'a pas été fermée (paramètre **ENDOPT** =*NONE ou *APP ou *DOC dans la dernière commande CPYTOMDB), la connexion précédente est de nouveau utilisée.

adresse IP ou nom de l'hôte

Le traitement est exécuté sur un autre poste que celui du demandeur. Dans ce cas, la saisie du nom du poste en question est demandé.

Remarque : Il est impératif que le programme LAUNCHER soit démarré sur le poste sur lequel les demandes sont exécutées.

Option de fin (ENDOPT)

Indique si la communication LAUNCHER Office sera fermée après traitement.

Les valeurs possibles sont :

***ALL**

La communication LAUNCHER entre le job et le serveur PC sera fermée.

***APP**

L'application Access sera fermée mais la communication LAUNCHER entre le job et le serveur PC restera ouverte.



L'application pourra envoyer une autre commande
CPYTOMDB en spécifiant valeur *CURRENT au paramètre EXESRV.

***DOC**

Le fichier Access sera fermé mais l'application et la communication
LAUNCHER entre le job et le serveur PC resteront actives.

***CON**

Seule la connexion LAUNCHER est fermée.

***NONE**

La communication LAUNCHER entre le job et le serveur PC ne sera pas
fermée ainsi que l'application, ACCESS et le fichier ACCESS.

Exemple

Description : COPIE D'UN FICHER PF AS/400 VERS UNE BASE MS ACCESS

La table nommée "TABLE1" est utilisé pour la copie.

A noter que le chemin d'accès à la base de données est indiqué au

paramètre **TOMDB** et qu'il ne doit pas être spécifié au paramètre **MDBPATH**

Si la base de données existe déjà, les données de la table sont remplacées.

Si la base de données n'existe pas, elle est automatiquement créée et la
table TABLE1 est ajoutée.

PGM

```
CPYTOMDB SELMDB (F) TOMDB ('%LNCDIR%\SAMPLES\R_MDB') +  
MDBPTH (*NONE) FROMFILE (SP_CUST) +  
TOTBL ('TABLE1') TBLOPT (*ADD) UNICOD (*YES) +  
EXESRV (*DEV) ENDOPT (*APP)
```

```
LNCShell CMD ('%LNCDIR%\Samples\S_LNCSQL.MDB') +  
VISIBLE (*YES) ACTION (OPEN) EXESRV (*CURRENT)
```

ENDPGM

CRTPRNSPLF - Commande CL

La commande **CRTPRNSPLF** permet de créer un fichier spoule AS/400 à partir d'un fichier de type PRN présent sur l'IFS de l'AS/400.

Un fichier PRN est un fichier qui peut être créé depuis toute application Windows en utilisant l'option 'Imprimer' et en cochant 'Imprimer dans un fichier'.

Ce fichier PRN peut également être obtenu par envoi de commandes LAUNCHER Office (par exemple **LNCPRNDOC**).

Ce fichier PRN peut ensuite être transféré sur l'IFS à l'aide de la commande **IFSPUT** de LAUNCHER Office.

Paramètres

| | | |
|--------------------------------|-----------------|---------------------------|
| Nom du fichier PRN sur l'IFS | FILE | |
| Nom du fichier spoule | SPLFNAME | Valeur alpha |
| Nom d'utilisateur | USER | Valeur alpha, *CURRENT |
| File d'attente en sortie | OUTQ | Nom, *JOB |
| Bibliothèque | *LIBL | Nom, *LIBL, *CURLIB |
| Type d'imprimé | FORMTYPE | Valeur alpha, *STD |
| Nombre d'exemplaires | COPIES | 1-255 |
| Suspendre l'impression | HOLD | *NO, *YES |
| Conserver l'impression . . | SAVE | *NO, *YES |
| Priorité de sortie (en OUTQ) . | OUTPTY | 1-9, *JOB |
| Référence utilisateur . | USRDTA | Valeur alpha |
| Texte utilisateur | | |
| Type de données imprimante | | *USERASCII |
| Rotation des pages | | *NO |

Détails

Nom du fichier PRN sur l'IFS (FILE)

Spécifie le nom du fichier PRN présent sur l'IFS et servant à la création du fichier spoule.

Ce paramètre est obligatoire.

Nom du fichier spoule (SPLFNAME)

Indique le nom du fichier spoule généré sur l'AS/400.

Les valeurs possibles sont :

LNCPRNPRT

Le fichier spoule **LNCPRNPRT** est utilisé par défaut.

nom-fichier-spole

Indiquez le nom du fichier spoule en sortie (10 caractères maximum).

Nom d'utilisateur (USER)

Indique le nom d'utilisateur sous lequel créer le spoule.

Les valeurs possibles sont :

***CURRENT**

L'utilisateur en cours est pris en considération.



nom-utilisateur

Indiquez un nom d'utilisateur As400 pour le fichier spoule généré.

File d'attente en sortie (OUTQ)

Indique le nom qualifié de la file d'attente en sortie.

Les valeurs possibles sont les suivantes :

***JOB**

La file d'attente en sortie associée au travail est utilisée pour la sortie spoule.

file-attente-sortie

Indiquez le nom de la file d'attente en sortie.

Les valeurs possibles pour la bibliothèque :

***LIBL**

La recherche porte sur toutes les bibliothèques de la liste des bibliothèques du travail jusqu'à ce que la première occurrence soit trouvée.

***CURLIB**

La file d'attente en sortie est recherchée dans la bibliothèque en cours du travail. Si celle-ci n'est pas précisée, QGPL est prise par défaut.

nom-bibliothèque

Indiquez le nom de la bibliothèque contenant la file d'attente en sortie.

Type d'imprimé (FORMTYPE)

Indique le type de papier utilisé par l'imprimante.

Les valeurs possibles sont :

***STD**

L'imprimé standard de votre système est utilisé.

type-imprimé

Indiquez l'identificateur du type de papier utilisé avec ce fichier unité pour les sorties imprimées.

Nombre d'exemplaires (COPIES)

Indique, le nombre d'exemplaires souhaité pour le fichier spoule généré.

Cette valeur doit être comprise entre **1** et **255**

Suspendre l'impression (HOLD)

Indique si le fichier spoule est suspendu ou non.

Les valeurs possibles sont :

***NO**

Le fichier spoule n'est pas suspendu en file d'attente de sortie.

***YES**

Le fichier spoule en sortie est suspendu jusqu'à ce qu'il soit libéré par la commande RLSSPLF(Libérer un fichier spoule).

Conserver l'impression (SAVE)

Indique si le fichier spoule généré est sauvegardé (conservé dans la file d'attente en sortie) une fois la sortie produite.

Les valeurs possibles sont :

***NO**

Le fichier spoule ne sera pas conservé dans la file d'attente de sortie après l'édition.

***YES**

Le fichier spoule sera conservé dans la file d'attente de sortie après l'édition.

Priorité de sortie (en OUTQ)

Indique la priorité de sortie pour les fichiers spoule en sortie qui sont produits par le travail. Le niveau de priorité le plus élevé est 1, et le niveau le plus bas est 9.

Les valeurs possibles sont :

***JOB**

La priorité de sortie du travail en cours est utilisée.

priorité-en-sortie

Indiquez un nombre de 1 (priorité la plus élevée) à 9 (priorité la plus faible) correspondant à la priorité de sortie du fichier spoule.

Référence utilisateur (USRDTA)

Indique la référence définie par l'utilisateur pour identifier le fichier spoule. Cette référence est un texte d'une longueur maximale de 10 caractères.

Texte utilisateur (USRTXT)

Représente un texte de 100 caractères au maximum.

Par exemple des informations de routage, peut être entré dans cette zone.

Type de données imprimante (DEVTYPE)

*USERASCII (par défaut)

*SCS

*LINE

*AFPDS

*AFPDSL

*IPDS

Rotation des pages

*YES

*NO (par défaut)

Exemple

L'exemple suivant imprime un document vers un fichier PRN, envoie le fichier PRN vers l'IFS, et le transfère ensuite vers un fichier spool.

```
LNCPRDOC      DOC('C:\Modeles\layout.doc')
               MRGTYPE(*FILE)
               FROMFILE(SP_LAYOUT)
               OUTPUT(*OUTPRN)
               OUTPRN('C:\temp\layout.prn')
               ENDOPT(*APP)

CHGVAR        VAR(&CMD) VALUE('IFSPUT')
CHGVAR        VAR(&PARM1) VALUE('C:\Temp\layout.prn')
CHGVAR        VAR(&PARM2) VALUE('/QDLS/IMAGES/Layout.prn')
CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
```

&PARM2 &RESULT)

CRTPRNSPLF FILE ('/QDLS/IMAGES/Layout.prn')
SPLFNAME (PRNSPLF)
USRTEXT ('Document Word')

RMVLNK OBJLNK ('/QDLS/IMAGES/Layout.prn')
CALL PGM (LNCCLDSE)



LNCCHK - Commande CL

LNCCHK permet de tester la connexion entre l'AS400 et le serveur LAUNCHER Office côté Windows.

Il vérifie également le lancement de Word et Excel, et affiche tous les versions des produits dans une fenêtre.

Cette commande ne marche pas avec Launcher en mode service Windows.

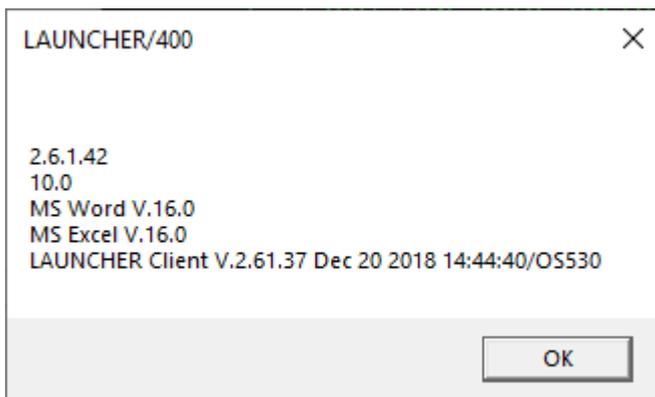
Paramètres

Nom du PC hébergeant le serveur LAUNCHER : ***DEV** par défaut

Exemple

```
LNCCHK EXESRV(*DEV)
```

Résultat obtenu:



LNCPRDOC - Commande CL

La commande **LNCPRDOC** permet de composer un document à partir d'un modèle Word, et des données d'un fichier ou d'une requête de l'AS/400.

La commande peut réaliser les opérations suivantes :

- Fusion du modèle et des données
- Sauvegarde du document résolu
- Impression du document résolu
- Affichage du document

Il est conseillé d'utiliser des modèles de documents Word (*.dot, *.dotx) .

L'aide en ligne disponible sur l'AS400 peut être plus complète pour la commande LNCPRDOC.

Paramètres

| | | |
|--------------------------------|-----------------|-----------------|
| Nom du modèle de Document | DOC | |
| Dossier / répertoire du modèle | FLR | *DFT |
| Racine du répertoire | ROOT | *DFT |
| Type source de données fusion | MRGTYPE | *NONE |
| Fichier d'origine | FROMFILE | |
| Bibliothèque | | *LIBL |
| Membre d'origine | FROMMBR | *FIRST |
| Requête SQL | FROMSQL | 'SELECT * FROM' |
| Analyse | FROMQRY | |
| Bibliothèque | | *LIBL |
| Type de sortie | OUTPUT | * |
| Sauvegarder document généré | OUTSAVE | *NO |
| Afficher le document généré | SHOWDOC | *NO |
| Nom du PC serveur LAUNCHER | EXESRV | *DFT |
| Option de fin | ENDOPT | *ALL |
| Répertoire par défaut de Word | WORDPATH | *DFT |
| Exécuter le publipostage | EXECMRG | *YES |
| Suppression des champs | NLINK | *YES |

Détails

La commande **LNCPRDOC** permet de fusionner un modèle de document WORD, avec des données provenant de l'AS/400.

Nom du modèle de document (DOC)

Indique le nom du modèle de document WORD à utiliser pour le publipostage.

Les valeurs possibles sont :

Nom du document

Indiquez le nom du document.

Remarque : Il est possible de spécifier le chemin d'accès complet dans ce paramètre.

Dans ce cas, les paramètres FLR et ROOT doivent avoir la valeur *NONE.

Un chemin réseau sous la forme :
\\Serveur\Partage\répertoire\document peut être spécifié.

***NEW**

Un nouveau document vierge est créé.
Cette option permet de préparer un nouveau modèle vierge, qui sera lié à une source de données, et qui sera ainsi prêt à être composé par l'utilisateur.

***CURRENT**

Cette valeur n'est possible, que si la communication LAUNCHER/400 est déjà ouverte entre le travail courant et le PC, et si un document est ouvert par un précédent appel à une commande de LAUNCHER/400. Voir le paramètre **ENDOPT**.

Dossier/répertoire du modèle (FLR)

Indique le nom du dossier ou répertoire, dans lequel se trouve le document.

Remarque : Ce paramètre est facultatif dans le cas où le paramètre **DOC** est renseigné avec le chemin complet

(D:\Racine\Répertoire\Document)

Le caractère '\' ne doit pas figurer à la fin du chemin saisi.

Racine du répertoire (ROOT)

Indique le chemin d'accès pour parvenir aux répertoires des modèles ou des documents résolus sauvegardés.

Remarque : Ce paramètre est facultatif. S'il est spécifié, ce paramètre est utilisé pour construire le chemin vers le modèle de document (DOC) et vers le document à sauvegarder (SAVDOC). Le caractère '\' ne doit pas figurer à la fin du chemin saisi.

Un chemin réseau sous la forme **\\Serveur\Partage\répertoire** peut être spécifié pour le paramètre ROOT.

Type source de données fusion (MRGTYPE)

Spécifie le type de source de données à utiliser pour extraire les données de l'AS/400 en vue de la fusion vers le document Word.

Les valeurs possibles sont :

***NONE**

Aucune fusion n'est faite. Le document sera affiché ou imprimé.

***QRY**

Les données à fusionner sont issues d'une requête AS/400 (objet de type *QRYDFN).

***FILE**

Les données à fusionner sont issues d'un fichier de base de données AS/400.

***SQL**



Les données à fusionner sont issues d'une requête SQL.

***DOC**

La source de données est définie au niveau du modèle de document Word.

Fichier d'origine (FROMFILE)

Indique le nom et la bibliothèque du fichier base de données à utiliser. Ce paramètre est utilisé lorsque le paramètre **MRGTYPE** est défini sur ***FILE**.

Le fichier base de données peut être physique ou logique.

Pour la bibliothèque, les valeurs possibles sont les suivantes :

***LIBL**

La recherche est effectuée dans toutes les bibliothèques utilisateur et système de la liste des bibliothèques du travail jusqu'à ce que la première occurrence soit trouvée.

***CURLIB**

Le fichier base de données est recherché dans la bibliothèque en cours du travail, si celle-ci n'est pas précisée.

Membre d'origine (FROMMBR)

Indique le nom du membre fichier base de données, à utiliser.

Ce paramètre est utilisé lorsque le paramètre **MRGTYPE** est défini sur ***FILE**.

Les valeurs possibles sont les suivantes :

***FIRST**

Le premier membre du fichier base de données est copié.

nom-membre

Indique le nom du membre du fichier base de données d'origine.

Requête SQL (FROMSQL)

Saisir la requête SQL pour extraire les données.

MRGTYPE = *SQL.

Analyse (FROMQRY)

Spécifie le nom de la requête à utiliser pour obtenir les données.

Ce paramètre est utilisé lorsque le paramètre **MRGTYPE** est défini sur ***QRY**.

Nom de la requête

Entrez le nom de la requête.

Les valeurs de bibliothèque possible sont:

*** LIBL**

La liste de bibliothèques est explorée pour trouver la bibliothèque où se trouve le fichier de requête.

Nom de la bibliothèque

Entrez le nom de la bibliothèque contenant le fichier de requête.

Type de sortie (OUTPUT)

Spécifie si le document généré sera imprimé, ou non.

Les valeurs possibles sont :

Le document ne sera pas imprimé. Il sera affiché si le paramètre **SHOWDOC** a la valeur ***YES**.

***PRINT**



Le document sera imprimé.

***OUTPRN**

Le document sera imprimé dans un fichier d'impression(PRN).

***PDF**

Le document sera converti en format PDF.

Le paramètre **OUTPDF** doit être renseigné avec le chemin et le nom du fichier PDF à générer.

***EMAIL**

Chaque lettre sera envoyée vers un destinataire de messagerie.

Renseignez le paramètre **MAILOPT** pour indiquer le sujet et le nom de la colonne du fichier contenant les adresse des destinataires.

Cette fonction n'est disponible que avec un client de messagerie OUTLOOK.

Fichier d'impression (OUTPRN)

Spécifie le chemin et le nom du fichier sur lequel imprimer, lorsque le paramètre **OUTPUT** est défini sur la valeur *OUTPRN.

Sauvegarder document généré (OUTSAVE)

Spécifie si le document résultant de la fusion doit être sauvegardé.

Les valeurs possibles sont :

***NO**

Le document résolu n'est pas sauvegardé.

***YES**

Le document résolu sera sauvegardé après la fusion. Le paramètre **SAVDOC** doit être renseigné.

Afficher le document (SHOWDOC)

Indique si le document est affiché à l'écran.

Les valeurs possibles sont :

***NO**

Le document résolu n'est pas montré à l'écran.

***YES**

Le document résolu sera rendu visible à l'écran.

Nom du serveur Launcher (EXESRV)

Spécifie le nom du serveur sur lequel WORD doit être exécuté. Ce nom peut correspondre à un autre poste de travail. Le nom du Host ou l'adresse IP sont des choix valides.

Les valeurs possibles sont :

***DFT**

Le traitement est exécuté sur le serveur référencé dans le fichier **LNCDFTP** réservé à la définition des options LAUNCHER Office (**mot clé LNCSRV**).

***DEV**

Le traitement est exécuté sur le poste de l'utilisateur interactif courant.

Cette valeur ne pas être utilisée dans un travail soumis.



***CURRENT**

Dans le cas où la dernière connexion n'a pas été fermée (paramètre ENDOPT=*NONE dans la dernière commande LNCPRDOC), la connexion précédente est de nouveau utilisée.

adresse IP ou nom de l'hôte

La traitement est exécuté sur un PC du réseau, désigné ici par son nom d'hôte, ou son adresse TCP/IP.

Remarque : Il est impératif que le programme LAUNCHER soit démarré sur le poste sur lequel les demandes sont exécutées.

Option de fin (ENDOPT)

Indique si la communication LAUNCHER Office sera fermée après traitement de la commande LNCPRDOC et précise quel élément doit être fermé.

Les valeurs possibles sont :

***ALL**

La communication LAUNCHER entre le job et le serveur PC sera fermée. L'application Word et le document sont aussi fermés.

***APP**

L'application sera fermée mais la communication LAUNCHER entre le job et le serveur P.C. restera ouverte.

***DOC**

Le document sera fermé mais l'application et la communication LAUNCHER entre le job et le serveur P.C. resteront actives.

***NONE**

La communication LAUNCHER entre le job et le serveur PC ne sera pas fermée ainsi que l'application et le document actif.

L'application pourra envoyer une autre commande

LNCPRDOC en spécifiant la valeur ***CURRENT** au paramètre **EXESRV**

Nom du document à enregistrer (SAVDOC)

Spécifie le nom du nouveau document à enregistrer après le publipostage.

Remarque: Ce paramètre peut contenir le chemin du répertoire complet.

L'extension du fichier doit être définie.

Exemple : MyResultLetter.docx

Répertoire de sauvegarde. (SAVFLR)

Spécifie le dossier ou le répertoire dans lequel le document résolu sera enregistré.

Les valeurs possibles sont:

***DFT**

La valeur par défaut du fichier LNCDFTP, pour le mot clé LNCSAV, sera utilisée.

Nom de dossier

Le chemin du dossier ou du répertoire est spécifié.

Remarque: le caractère '\' ne doit pas être saisi à la fin de la chaîne.

***NONE**

Spécifie que le paramètre **SAVDOC** contient le chemin du répertoire.

Type de fusion Word (MRGSEL)

Spécifie le type de fusion que Word utilisera.

Les valeurs possibles sont:

***LTR**

Le mot fusion est une lettre.

Une lettre est créée pour chaque enregistrement de la source de données.
Les lettres sont séparées les unes des autres par un "Form Feed".

***CAT**

La fusion de mots est un catalogue.

Un seul document est créé, avec tous les enregistrements de la source de données.

***NONE**

Le type de fusion est défini dans le modèle Word.

Répertoire par défaut de Word (WORDPATH)

Indique le chemin du répertoire utilisé par Word pour inclure des documents ou des objets externes.

Les valeurs possibles sont :

***DFT**

Le répertoire utilisé par défaut par l'application WORD fait référence au mot clé **LNCDIR** référencé dans le fichier **LNCDFTP**.

répertoire_par_défaut

Indiquer le nom du répertoire utilisé par WORD par défaut.

Remarque : Le nom complet du chemin d'accès (**Root/Dossier**) doit être spécifié.

***NONE**

Si *NONE est spécifié, le répertoire WORD utilisé par défaut correspond au répertoire défini dans les options Word.

Fichier de transfert temporaire (DOCTMP)

Les données sont transférées dans un fichier temporaire sur le PC.

Le nom de fichier et le chemin d'accès à ce fichier peuvent être définis.

Valeurs possibles :

***DFT**

La valeur du mot clé LNCTMP dans le fichier LNCDFTP est utilisée.

nom fichier temporaire

Définit le chemin et le nom du fichier temporaire à utiliser.

Word merge options (WORDMRG))

Several options are available.

Header file

Specifies the name of a file containing the fields names for the data source.

Possible values are :

***FILE**

The fields names come from the AS/400 database file.

Header-file-name

Enter a PC file name containing the names of the fields.

Suppress blank lines

In the document, when a line containing merge fields is blank after the mail merge, Word can suppress it.

Possible values are :

***NO**

The blank lines are kept in the resolved document.

***YES**

The blank lines are deleted from the resolved document.

Exécuter le publipostage (EXECMRG)



Indique si le publipostage doit être effectué ou seulement préparé.

Les valeurs possibles sont :

***YES**

Le publipostage est exécuté.

***NO**

Le publipostage n'est pas exécuté.

Les liaisons avec la source de données et d'entête sont préparées.

Le modèle peut être complété sous Word.

Exemple

```
LNCPRDOC SELDOC(F) DOC('%LNCDIR%\Samples\sp_cust.docx') FLR(*NONE)
ROOT(*NONE) MRGTYPE(*FILE) FROMFILE(SP_CUST) SHOWDOC(*YES)
EXESRV(*DEV)
```

LNCHELL - Commande CL

La commande **LNCHELL** exécute une commande DOS, ouvre un document ou exécute un programme sur une station Windows.

Paramètres

| | | |
|--------------------------------------|------------------|-------|
| Commande, Document ou Exe | CDM | |
| Répertoire du Doc ou Exe | FLR | *NONE |
| Paramètres du prog. Exécutable | PARMEXE | *NONE |
| Attendre fin exécution | WAITCMD | *NO |
| Retourner niveau d'erreur | ERRORLVL | *YES |
| Exécution visible | VISIBLE | *NO |
| Placer la fenêtre en 1er plan | FOCUS | *YES |
| Minimiser la fenêtre active | MINCURWIN | *NO |
| Action sur le document ou Exe | ACTION | *NONE |
| Répertoire par défaut | WRKFLR | *NONE |
| Nom du PC serveur LAUNCHER | EXESRV | *DFT |
| Option de fermeture de communication | ENDOPT | *ALL |

Détails

Commande, Document ou Exe (CMD)

Indique : Une commande DOS à exécuter, un nom de document à ouvrir, ou un nom de programme exécutable.

Répertoire du Doc. ou Exe (FLR)

Si le paramètre **CMD** désigne un document ou un programme exécutable, **FLR** peut alors désigner un chemin de répertoire où se trouve le fichier.
Remarque : Ce paramètre est facultatif dans le cas où le paramètre **CMD** est renseigné avec le chemin complet (**D:\Racine\Répertoire\Document**)
 Le caractère '\' ne doit pas figurer à la fin du chemin saisi.
 Un chemin réseau sous la forme **\\Serveur\Partage\répertoire** peut être spécifié pour le paramètre **FLR**.

Paramètres du prog. Exécutable (PARMEXE)

Si le paramètre **CMD** désigne un programme exécutable, des paramètres à passer peuvent être indiqués ici.

Attendre fin exécution (WAITCMD)

Les valeurs possibles sont :

***NO**

L'application n'attend pas que la commande soit exécutée.

***YES**

L'application attend indéfiniment que la commande soit exécutée.

La main sera rendu au programme AS/400 quand l'exécution de la commande sera terminée.

Valeur

Le travail attend la fermeture du programme appelé sur le PC, jusqu'à un délai maximum exprimé en millisecondes. Si le délai est expiré, le message



LNC0701 est envoyé au travail, avec la valeur **99999** dans les 5 premiers caractères.

Attention, certaines applications, tel que Internet Explorer, ne permettent pas d'attendre la fin d'exécution.

Retour niveau d'erreur (ERRORLVL)

Indique si le retour de l'appel de commande doit être testé.

Cette option n'est valide que si **WAITCMD** a la valeur ***YES**.

Les valeurs possibles sont :

***YES**

Si un code différent de 0 est retourné, le message **LNC0701** est envoyé au travail.

Les 5 premiers caractères du message représentent le numéro d'erreur.

La valeur 99999 est retournée quand le délai exprimé par **WAITCMD** a été dépassé.

***NO**

Le retour de l'appel de commande n'est pas testé.

Exécution visible (VISIBLE)

Les valeurs possibles sont :

***NO**

L'exécution de la commande n'est pas visible.

***YES**

L'exécution de la commande est visible.

Placer la fenêtre en 1^{er} plan (FOCUS)

Les valeurs possibles sont :

***YES**

La fenêtre de l'application lancée sur le PC sera placée au 1er plan des fenêtres actives sur Windows.

***NO**

La fenêtre active restera inchangée.

Ceci permet de garder la fenêtre de l'émulation de terminal toujours active.

Minimiser la fenêtre active (MINCURWIN)

Les valeurs possibles sont :

***YES**

La fenêtre active est minimisée.

Si **WAITCMD** n'a pas la valeur ***NO**, la fenêtre sera restaurée à la fin de l'exécution du programme appelé.

***NO**

La fenêtre active reste affichée dans sa taille normale.

Action sur le document ou Exe (ACTION)

Ce paramètre ne s'applique pas à l'exécution de commandes DOS (paramètre **CMD**).

Les valeurs possibles sont :

OPEN

Cette valeur doit être indiquée si le paramètre **CMD** désigne un programme exécutable lorsque le paramètre **CMD** désigne un document, la valeur **OPEN** permet d'ouvrir le document sous Windows, avec son application par défaut.

PRINT

Lorsque le paramètre **CMD** désigne un document, la valeur **PRINT** permet d'imprimer le document sous Windows.

Répertoire par défaut (WRKFLR)

Désigne le répertoire par défaut pendant l'exécution de la commande ou du programme.

Nom du serveur LAUNCHER (EXESRV)

Spécifie le nom du serveur sur lequel la commande doit être exécutée.

Ce nom peut correspondre à un autre poste de travail.

Le nom du Host ou l'adresse IP sont des choix valides.

Les valeurs possibles sont :

***DFT**

Le traitement est exécuté sur le serveur référencé dans le fichier **LNCDFTP** réservé à la définition des options LAUNCHER Office (mot clé **LNCSRV**).

***DEV**

Le traitement est exécuté sur le poste de l'utilisateur interactif courant. Cette valeur ne pas être utilisée dans un travail soumis.

***CURRENT**

Dans le cas où la dernière connexion n'a pas été fermée (paramètre ENDOPT=*NO) lors de la dernière commande LNCSHELL, la connexion précédente est de nouveau utilisée.

adresse IP ou nom de l'hôte

Le traitement est exécuté sur un PC du réseau, désigné ici par son nom d'hôte, ou son adresse TCP/IP.

Remarque : Il est impératif que le programme LAUNCHER Office soit démarré sur le poste sur lequel les demandes sont exécutées.

Option de fermeture de comm (ENDOPT)

Indique si la communication est fermée après le traitement de la commande.

Les valeurs possibles sont :

***ALL**

La communication LAUNCHER entre le job et le serveur PC sera fermée.
L'application ne pourra pas envoyer une autre commande en spécifiant
valeur *CURRENT au paramètre **EXESRV**

***NONE**

La communication LAUNCHER entre le job et le serveur PC ne sera pas
fermée.
L'application pourra envoyer une autre commande en spécifiant la valeur
***CURRENT** au paramètre **EXESRV**.

Exemple

Création et ouverture d'un fichier PDF :

```
PGM
```

```
LNCOPEN
```

```
LNCCMD CMD(WORDOPEN) PARM1(NEW)
```

```
LNCCMD CMD(WTYPETEXT) PARM1('CECI EST UN FICHIER QUI A ETE +  
CONVERTI EN FICHIER PDF')
```

```
LNCCMD CMD(PDFPRINTER) +  
PARM1('FILE="C:\TEMP\TEMP.PDF"')
```

```
LNCCMD CMD(WPRINT) PARM1('PRINTER="LAUNCHER_PDF"')
```

```
LNCCMD CMD(WORDCLOSE)
```

```
LNCSHELL CMD('C:\TEMP\TEMP.PDF') WAITCMD(*NO) VISIBLE(*YES) +  
ACTION(OPEN) EXESRV(*DEV)
```

```
LNCCLOSE
```

```
ENDPGM
```

LNCSDMAIL - Commande CL

La commande **LNCSDMAIL** permet d'envoyer des messages électroniques et d'en contrôler les principales caractéristiques :
 les destinataires (direct, en copie) l'objet, le texte du message, la (les) pièce(s) jointe(s), le niveau de priorité, l'activation de l'accusé de réception.
 Les messageries utilisées doivent être compatibles MAPI (ce qui est le cas de Outlook ou Exchange), sinon vous pouvez également utiliser Lotus Notes ou encore utiliser un serveur SMTP.

Remarque : Pour envoyer vos messages, il n'est pas nécessaire que le système de messagerie soit en cours d'exécution sur le poste.

Paramètres

| | | |
|---|-----------------|---------|
| Pièce jointe + si autres valeurs | DOC | *NONE |
| Dossier des pièces jointes | FLR | *NONE |
| Destinataire (s) | TOINTNET | |
| Adresse de messagerie | | |
| Type de destinataire + si autres valeurs | | *TO |
| Sujet | SUBJECT | |
| Message | MSG | *NONE |
| Priorité | PTY | *NORMAL |
| Confirmation de réception | CFMDEL | *NO |
| Nom du PC serveur LAUNCHER | EXESRV | *DFT |
| Option de fin | | *ALL |
| Type de message utilisée | MSGTYPE | *SMTP |
| Paramètres SMTP : | | |
| Adresse email de l'expéditeur | CFGSMTP | *DFT |
| Adresse email de retour | | *DFT |
| Adresse ou nom du serveur SMTP | | *DFT |
| Port du serveur SMTP | | *DFT |
| Trace des envois effectués | | *NO |
| Profil utilisateur SMTP | | *NONE |
| Mot de passe de l'utilisateur | | *NONE |
| Activer SSL | | *NO |
| Timeout | | *NONE |
| Nombre essais | | *NONE |
| Chemin logfile | | *DFT |
| Protocole SSL | | TLS |
| Signature | | *NO |
| Chiffrer (encrypter) email | | *NO |
| Chemin certificat | | *NONE |
| Mot de passe certificat | | *NONE |
| Sécuriser les headers | | *NO |

| | |
|-------------------------|-------|
| Sujet du wrapper | *NONE |
| Vérifier adresses email | *NO |

Détails

Pièce jointe (DOC)

Nom et chemin de chaque fichier à attacher.
Ce paramètre autorise plusieurs valeurs.
Pour spécifier plusieurs fichiers, entrez un signe '+' dans la ligne "+ si autres valeurs" et tapez ENTER.

Dossier des pièces jointes (FLR)

Ce paramètre indique le nom du dossier pour les documents envoyés.
Si le chemin complet est indiqué au paramètre **DOC**, il faut mettre la valeur ***NONE** dans ce paramètre.

Destinataire(s) (TOINTNET)

Indique l'adresse de messagerie du ou des utilisateurs auxquels la distribution est envoyée.

Vous pouvez indiquer jusqu'à 100 adresses de messagerie.

Ce paramètre admet des valeurs multiples.

Pour obtenir des zones d'entrée supplémentaires à l'écran, entrez un signe plus (+) en regard de l'invite "+ si autres valeurs", puis appuyez sur ENTREE.

adresse de messagerie

Indiquez l'adresse de messagerie de la personne ou de l'entreprise à laquelle la distribution est envoyée (jusqu'à 253 caractères par adresse).

type de destinataire

Indique le type de destinataire utilisé.

Les valeurs possibles de type de destinataire sont les suivantes :

***TO**

L'adresse de messagerie correspond au destinataire principal de la distribution.

***CC**

L'adresse de messagerie recevra une copie de la distribution envoyée au destinataire principal.

***BCC**

L'adresse de messagerie recevra une copie de la distribution.

Cependant, le destinataire de la copie ne sera pas désigné sur la distribution comme étant destinataire de la distribution.

Sujet (SUBJECT)

Description de la distribution. Vous pouvez indiquer jusqu'à 44 caractères.

Ce paramètre est obligatoire et ne peut pas commencer par un blanc ni être constitué uniquement de blancs.

Message (MSG)

Indique le texte du message à envoyer aux destinataires.

Les valeurs possibles sont :

message



Saisir un message d'une longueur de 5000 caractères maximum.
Pour effectuer des sauts de lignes, taper **%PARA%** entre le texte de fin d'une ligne et le début de texte de la ligne suivante.

***NONE**

La valeur *NONE indique que l'envoi du mail ne contient pas de message.

***DOCMMSG**

La valeur *DOCMMSG permet à l'utilisateur d'envoyer un document contenu dans le corps du message.

Ce fichier peut être dans différents formats (texte, HTML, RTF).

Remarque : la valeur *DOCMMSG n'est admise que pour la messagerie SMTP.

Priorité (PTY)

Indique le niveau de priorité de la distribution (faible, normal ou élevé).

Les valeurs possibles sont les suivantes :

***NORMAL**

La distribution n'est pas prioritaire.

***HIGH**

La distribution est prioritaire.

***LOW**

La distribution est de faible priorité.

Confirmation de réception (CFMDEL)

Indique si l'expéditeur désire recevoir une confirmation de réception lorsque le destinataire reçoit la distribution.

Les valeurs possibles sont les suivantes :

***NO**

L'expéditeur ne demande pas de confirmation de réception.

***YES**

Une confirmation de réception est demandée.

Nom du serveur MAIL (EXESRV)

Spécifie le nom du serveur sur lequel l'envoi doit être exécuté.

Ce nom peut correspondre à un autre poste de travail.

Le nom du Host ou l'adresse IP sont des choix valides.

Les valeurs possibles sont :

***DFT**

Le traitement est exécuté sur le serveur référencé dans le fichier LNCDFTP réservé à la définition des options LAUNCHER Office (mot clé MSGSRV).

***DEV**

Le traitement est exécuté sur le poste de l'utilisateur interactif courant.

Cette valeur ne pas être utilisée dans un travail soumis.

adresse IP ou nom de l'hote

Le traitement est exécuté sur un PC du réseau, désigné ici par son nom d'hote, ou son adresse TCP/IP.

Remarque : Il est impératif que le programme LAUNCHER Office soit démarré sur le poste sur lequel les demandes sont exécutées.

Type de messagerie utilisé (MSGTYPE)

Ce paramètre indique le type de messagerie utilisé pour l'envoi du message ou du document vers le destinataire final.



***SMTP**

La messagerie utilisée est de type SMTP.

***MAPI**

La messagerie utilisée est de type MAPI.

***LOTUS**

La messagerie utilisée est de type LOTUS.

Paramètres SMTP (CFGSMTP)

Ces paramètres sont requis dans le cas d'une messagerie SMTP.

Adresse email de l'expéditeur

Indique l'adresse email utilisé pour l'envoi des messages.

Les valeurs possibles sont :

***DFT**

L'envoi utilise l'adresse email spécifié dans le mot clé MSGSND présent dans le fichier LNCDFTP.

adresse_email

Spécifiez l'adresse email de la personne qui envoi la distribution.

Adresse email de retour

Spécifie l'adresse de la personne qui réceptionne les réponses aux messages envoyés.

Adresse ou nom du serveur SMTP

Spécifie l'adresse IP du serveur SMTP utilisée.

Port du serveur SMTP

Port du serveur SMTP dans le cas ou le serveur n'utilise pas le port 25.

Trace des envois effectués

Valeur par défaut : ***NO**.

***YES** : Permet de s'assurer que l'envoi du message s'est déroulé sans erreur.

Ce fichier est formaté par des séparateurs (;), pour pouvoir être importé dans une base de données ou EXCEL.

Utilisez WRKLNCDFT pour spécifier le chemin et le nom du fichier logfile.

Profil utilisateur SMTP

Profil utilisateur SMTP.

Mot de passe utilisateur SMTP

Mot de passe de l'utilisateur SMTP.

Activer SSL

Valeur par défaut : ***NO**.

SSL est utilisé pour accéder au serveur de messagerie SMTP spécifié.

Le protocole TLS est utilisé par défaut. Changez le paramètre Protocole SSL si vous voulez utiliser le protocole SMTPS.

Timeout

Définit le temps d'attente pour l'envoi d'un e-mail. Par défaut, aucun délai n'est spécifié (***NONE**). C'est le temps de traitement du serveur SMTP une fois la connexion établie. Indiquez un délai en secondes.

Nombre essais

Indiquez le nombre d'essais pour envoyer un e-mail si le serveur SMTP n'est pas accessible. Par défaut, aucune nouvelle tentative (***NONE**).

Indiquez un nombre de nouvelles tentatives.

Chemin logfile

Si la trace des envois effectués est activée (***YES**), par défaut le chemin du logfile (***DFT**) est défini avec la commande WRKLNCDFT. Vous pouvez spécifier ici un autre chemin que ***DFT**.

Protocole SSL

Par défaut: le protocole TLS (commande STARTTLS) est utilisé.
Le port #587 est généralement utilisé avec le protocole TLS.
Vous pouvez spécifier le protocole SMTPS, également appelé SMTP/SSL,
SMTP over SSL ou SMTPS et utilise généralement le port #465.

Signer email

Mettre *YES pour signer l'email. Le chemin et le mot de passe du certificat sont obligatoires. Par défaut : *NO.

Chiffrer (encrypter) email

Le corps du mail est chiffré selon PKCS #7. Le chemin et le mot de passe du certificat sont obligatoires. Par défaut : *NO.

Chemin certificat

Chemin du certificat(fichier .p12) utilisé pour signer et/ou chiffrer l'email.

Mot de passe certificat

Mot de passe du certificat(fichier .p12) utilisé pour signer et/ou chiffrer l'email.

Sécuriser headers

Protection des en-têtes grâce à l'utilisation d'un message RFC 822. Un wrapper message/rfc822 est utilisé pour appliquer les services de sécurité S/MIME aux headers.

Voir norme RFC 8551 pour plus de détails. L'email sera reçu en pièce jointe par le destinataire.

Sujet du wrapper

Si SecureHeaders=*YES, par défaut le destinataire recevra un email sans objet et une pièce jointe contenant l'email spécifié. Vous pouvez spécifier ici un sujet afin d'éviter que l'email aille dans les spams.

Vérifier adresses email

Mettre *YES pour effectuer une vérification des adresses email des destinataires selon la norme RFC 5322, avant envoi. Par défaut : *NO.

Exemple

```
LNCSDMAIL TOINTNET(('tech@easycom-aura.com' *CC) ('INFO@EASYCOM-AURA.COM')) SUBJECT('ENVOI DE MAIL') MSG('Ceci est un test d'envoi de mail avec LAUNCHER Office') CFGSMTP('TECH@EASYCOM-AURA.COM' 'TECH@EASYCOM-AURA.COM' '183.145.89.12')
```

Autre exemple

Envoi d'un fichier HTML en corps de mail avec SMTP. Accusé de réception demandé.

Enregistrement des envois activé.

```
LNCSDMAIL TOINTNET(('tech@easycom-aura.com') +  
('sales@easycom-aura.com' *BCC)) +  
SUBJECT(&OBJET) MSG(*DOCMSG) CFMDEL(*YES) +  
EXESRV(*CURRENT) +  
CFGSMTP('info@easycom-aura.com' +  
'info@easycom-aura.com' +  
'smtp.aura.fr' 0 *DFT *YES) +  
DOCMSG('C:\A\R_11WORD.HTML')
```



LNCTOXLS - Commande CL

La commande **LNCTOXLS** permet de transférer des données de l'AS/400 dans une feuille Excel.

Paramètres

| | | |
|--------------------------------|-----------------|-------------------------------|
| Nom du Classeur modèle Excel | TOXLS | |
| Dossier/répertoire du classeur | XLSPATH | *DFT |
| Type de la source de données | CPYSRC | *FILE, *QRY, *SQL, *NONE |
| Nom du fichier base de données | FROMFILE | |
| Bibliothèque | | Nom, *LIBL, *CURLIB |
| Membre du fichier | FROMMBR | Nom, *FIRST |
| Colonne du fichier | COLUMNS | *ALL |
| Conversion de types de données | TYPECAST | *NONE |
| Nom de la feuille EXCEL | TOSHEET | *ONLY |
| Nom plage de destination | TONAME | *NONE |
| Remplace/ajoute/Insere lignes | XLSOPT | *REPLACE, *ADD, *INSERT |
| Type de sortie | OUTPUT | *, *PRINT |
| Nom du document sauvegardé | SAVDOC | *NONE |
| Afficher le document | | |
| Nom du serveur LAUNCHER | | |
| Option de fin | SHOWDOC | *YES, *NO |
| Ajuster largeur des colonnes | EXESRV | *DFT |
| Appliquer Format aux cellules | ENDOPT | *ALL, *APP, *DOC, *CON, *NONE |
| Insérer entêtes de colonnes | AUTOFIT | *YES, *NO |
| Appliquer Format automatique | FMTCELLS | *YES, *NO |
| | ADDCOLH | *NONE, *COLHDG, *NAME, ALIAS |
| | AUTOFMT | *NONE |

Détails

Nom du classeur Excel (TOXLS)

Indique le nom du classeur EXCEL utilisé comme modèle.

Ce paramètre est obligatoire.

Les valeurs possibles sont :

nom-du-tableur-Excel

Indique le nom du classeur Excel utilisé en entrée.

***CURRENT**

Indique que le classeur Excel utilisé est le dernier ouvert par Launcher.

***NEW**

Indique que le classeur Excel utilisé est un nouveau classeur vierge.

Chemin d'accès au classeur (XLSPATH)

Indique le chemin d'accès pour accéder au classeur Excel.

Les valeurs possibles sont les suivantes :

***DFT**

La valeur spéciale ***DFT** indique que le chemin d'accès utilisé pour l'accès au



classeur Excel, est le chemin d'accès référencé dans le fichier des options par défaut Launcher400. Pour visualiser ou modifier la valeur du chemin d'accès utilisé par défaut par la commande LNCTOXLS, sélectionnez dans la liste le mot clé : **XLSPTH** (commande **WRKLNCDFT**).

***NONE**

Indique que le chemin d'accès est déjà indiqué au paramètre **TOXLS**.

nom-chemin-d'accès

Indiquez le nom complet du chemin d'accès utilisé pour accéder au classeur Excel.

Source de la copie (CPYSRC)

Indique l'origine de la source à copier.

Les valeurs possibles sont les suivantes :

***FILE**

L'origine de la copie est un fichier base de données.

***QRY**

L'origine de la copie est une analyse query (objet de type *QRYDFN).

***SQL**

L'origine de la copie est une instruction SQL **SELECT**

Fichier d'origine (FROMFILE)

Indique le nom et la bibliothèque du fichier base de données à utiliser (**CPYSRC = *FILE**).

Le fichier base de données peut-être physique ou logique.

Pour la bibliothèque, les valeurs possibles sont les suivantes :

***LIBL**

La recherche est effectuée dans toutes les bibliothèques utilisateur et système de la liste des bibliothèques du travail jusqu'à ce que la première occurrence soit trouvée.

***CURLIB**

Le fichier base de données est recherché dans la bibliothèque en cours du travail, si celle-ci n'est pas précisée.

Nom de la bibliothèque

Membre d'origine (FROMMBR)

Indique le nom du membre fichier base de données, à utiliser pour la copie.

Les valeurs possibles sont les suivantes :

***FIRST**

Le premier membre du fichier base de données est copié.

nom-membre

Indique le nom du membre du fichier base de données d'origine.

Requête SQL (FROMSQL)

Saisir la requête SQL pour extraire les données.

MRGTYPE = *SQL.

Analyse (FROMQRY)

Spécifie le nom de la requête à utiliser pour obtenir les données.

Ce paramètre est utilisé lorsque le paramètre **MRGTYPE** est défini sur ***QRY**.

Nom de la requête



Entrez le nom de la requête.

Les valeurs de bibliothèque possible sont:

*** LIBL**

La liste de bibliothèques est explorée pour trouver la bibliothèque où se trouve le fichier de requête.

Nom de la bibliothèque

Entrez le nom de la bibliothèque contenant le fichier de requête.

Colonnes du fichier (COLUMNS)

Permet de sélectionner les colonnes du fichier source qui seront envoyées vers le classeur.

Indiquez les noms des colonnes, séparés par le caractère virgule (,).

La valeur spéciale ***ALL** indique que toutes les colonnes du fichier source doivent être transférées vers le classeur.

Conversion de types de données (TYPECAST)

Permet de convertir le format de colonnes du fichier source.

Pour chaque colonne dont il faut changer le format, indiquez le nom de la colonne, suivi du caractère ':', puis du nouveau type de donnée à appliquer.

Les descriptions de conversion sont séparées par le caractère ','.

Exemple: **DATEO:DATE,PRODID:CHAR**

La colonne DATEO sera considérée comme une date, la colonne PRODID sera traitée comme caractère.

La valeur spéciale ***NONE** indique qu'aucune conversion de format n'est à effectuer.

Les valeurs possibles pour le nouveau type de donnée sont les suivantes :

DATE

La colonne dans la source de données doit contenir une valeur numérique exprimant une date dans le format AAAAMMJJ, ou AAMMJJ. La valeur numérique sera convertie en date.

CHAR

La valeur de la colonne sera considérée comme chaîne de caractères. Les zéros non significatifs sont conservés à gauche.

Nom de la feuille Excel (TOSHEET)

Indique le nom de la feuille Excel utilisée pour la fusion des données.

Les valeurs possibles sont les suivantes :

***ONLY**

La valeur spéciale ***ONLY** indique que la feuille du classeur utilisée pour la fusion des données As400 est la première feuille disponible et active après l'ouverture du classeur Excel.

Remarque : il est recommandé de n'utiliser cette option que lorsque le classeur Excel ne contient qu'une seule feuille.

***AUTOCRT**

La valeur spéciale ***AUTOCRT** indique qu'une feuille Excel est insérée automatiquement dans le classeur.

Cette feuille sera donc la destination de la copie des données provenant de l'AS/400.

nom-de-la-feuille-Excel



Indique que le nom de la feuille Excel dans laquelle sont importées les données.

Nom du tableau Excel (TONAME)

Indique un nom de plage créé dans une feuille Excel ou une référence à atteindre sous la forme de **\$Lettre_de_colonne\$numero_de_ligne**.

Exemple : TONAME(\$D\$3) représente la cellule : colonne 4 et ligne 3.

Remplacement/ajout des données (XLSOPT)

Indique comment les données sont transférées dans EXCEL.

Les valeurs possibles sont les suivantes :

***REPLACE**

Les nouvelles données remplacent les anciennes données.

***ADD**

Les nouvelles données sont ajoutées aux anciennes données.

***INSERT**

Des lignes sont insérées pour recevoir les nouvelles données.

Le paramètre **ENTIRER** détermine si des lignes entières sont insérées, ou non.

***RESIZE**

Les nouvelles données remplacent les anciennes données.

La zone dans le classeur est redimensionnée.

Des lignes seront supprimées ou insérées.

***MAP**

Les valeurs des colonnes de la source de données sont affectées aux cellules portant les mêmes noms dans le tableau Excel.

Seul le premier enregistrement de la source de donnée est utilisé.

Unité de sortie (OUTPUT)

Indique si la sortie générée par la commande doit être affichée sur le poste de travail demandeur, ou imprimée avec la sortie spoule du travail.

Les valeurs possibles sont :

Le Classeur EXCEL résolu s'affiche à l'écran.

***PRINT**

Le Classeur EXCEL est imprimé sur l'imprimante défini par défaut sur le poste de travail qui exécute la requête Launcher400.

Nom du document Excel sauveg. (SAVDOC)

Indique le nom du document Excel généré en sortie après la fusion des données.

Remarque : Si le chemin d'accès est omis, le document Excel sera sauvegardé dans le répertoire défini par défaut comme répertoire de sauvegarde.

Les valeurs possibles sont les suivantes :

***NONE**

Le document traité n'est pas sauvegardé après l'exécution de la commande

LNCTOxls.

Ceci peut-être valable dans le cas ou d'autres commandes **LNCTOxls** sont exécutés à la suite.

La dernière commande LNCTOxls doit dans ce cas indiquer le nom du



document sauvegardé.

fichier-Excel-généré-en-sortie

Indique le nom du document Excel pour sauvegarder les données traitées.

Répertoire de sauvegarde. (SAVFLR)

Spécifie le dossier ou le répertoire dans lequel le document final sera enregistré.

Les valeurs possibles sont:

***DFT**

La valeur par défaut du fichier LNCDFTP, pour le mot clé **XLSSAV**, sera utilisée.

Nom de dossier

Le chemin du dossier ou du répertoire est spécifié.

Remarque: le caractère '\' ne doit pas être saisi à la fin de la chaîne.

***NONE**

Spécifie que le paramètre **SAVDOC** contient le chemin du répertoire.

Format de sauvegarde. (SAVFMT)

Spécifie le format du classeur à sauvegarder.

Afficher le document (SHOWDOC)

Indique si le document est affiché ou non sur l'écran.

Les valeurs possibles sont :

***NO**

Le document résolu n'est pas montré à l'écran.

***YES**

Le document résolu sera rendu visible à l'écran.

Nom du serveur Launcher (EXESRV)

Spécifie le nom du serveur sur lequel Excel doit être exécuté. Ce nom peut correspondre à un autre poste de travail. Le nom du Host ou l'adresse IP sont des choix valides.

Les valeurs possibles sont :

***DFT**

Le traitement est exécuté sur le serveur référencé dans le fichier **LNCDFTP** réservé à la définition des options LAUNCHER Office (**mot clé XLSSRV**).

***DEV**

Le traitement est exécuté sur le poste de l'utilisateur interactif courant.

Cette valeur ne pas être utilisée dans un travail soumis.

***CURRENT**

Dans le cas où la dernière connexion n'a pas été fermée (paramètre ENDOPT=*NONE dans la dernière commande LNCTOxls), la connexion précédente est de nouveau utilisée.

adresse IP ou nom de l'hôte

La traitement est exécuté sur un PC du réseau, désigné ici par son nom d'hôte, ou son adresse TCP/IP.

Remarque : Il est impératif que le programme LAUNCHER soit démarré sur le poste sur lequel les demandes sont exécutées.

Option de fin (ENDOPT)

Indique si la communication LAUNCHER Office sera fermée après traitement de la commande LNCTOXLS et précise quel élément doit être fermé.

Les valeurs possibles sont :

***ALL**

La communication LAUNCHER entre le job et le serveur PC sera fermée. L'application Excel et le document sont aussi fermés.

***APP**

L'application sera fermée mais la communication LAUNCHER entre le job et le serveur P.C. restera ouverte.

***DOC**

Le document sera fermé mais l'application et la communication LAUNCHER entre le job et le serveur P.C. resteront actives.

***CON**

La communication LAUNCHER entre le job et le serveur PC est fermée.

***NONE**

La communication LAUNCHER entre le job et le serveur PC ne sera pas fermée ainsi que l'application et le document actif.

L'application pourra envoyer une autre commande

LNCTOXLS en spécifiant la valeur ***CURRENT** au paramètre **EXESRV**.

Ajuster la largeur des colonnes (AUTOFIT)

Indique si la largeur des colonnes doit être automatiquement ajustée en fonction de leur contenu.

Les valeurs possibles sont :

***YES**

Ajuster automatiquement la largeur des colonnes.

***NO**

Ne pas modifier la largeur des colonnes.

Appliquer un format automatique (AUTOFMT)

Indique si un format automatique doit être appliqué à la zone mise à jour avec les données de la base de données.

Les valeurs possibles sont :

***NONE**

Aucun format automatique n'est appliqué.

Nom

Le format automatique est appliqué.

Consultez la documentation technique de Excel pour connaître les noms des formats automatiques existants.

Fixer le format des cellules (FMTCELLS)

Indique si le format des cellules doit être fixé en fonction des types de données de la base de données.

Les valeurs possibles sont :

***YES**

Le format des cellules est fixé en fonction du type Base de Données.

***NO**

le format des cellules est fixé par Excel, en fonction des valeurs.



Insérer les entêtes de colonnes (ADDCOLH)

Indique si les noms de colonnes doivent être insérés en première ligne.

Les valeurs possibles sont :

***NONE**

Les noms de colonnes ne sont pas insérés.

***NAME**

Les noms des champs de la base de donnée sont insérés.

***COLHDG**

Les entêtes de colonnes de la base de donnée sont insérés.

Cette option fonctionne uniquement sur un fichier et pas sur une requête SQL.

***ALIAS**

Les alias (noms longs) des colonnes sont insérés.

Insérer des lignes entières (ENTIRER)

Si **XLSOPT** est défini sur la valeur ***INSERT**, cette option indique si des lignes entières sont insérées ou non.

Les valeurs possibles sont:

***YES**

Des lignes entières sont insérées pour recevoir les nouvelles données.

***NO**

Les cellules situées derrière la plage sélectionnée (le paramètre **TONAME**) sont déplacées.

Exemple

```
LNCTOxls TOxls (MODELE.XLSX) +
XLSPTH ('%LNCDIR%\SAMPLES') CPYSRC (*SQL) +
FROMSQL ('SELECT * FROM SP_ORD') +
TOSHEET (FEUIL1) +
TONAME ($A$2) XLSOPT (*ADD) +
SAVDOC (R_RESULTATS) +
SAVFLR ('%LNCDIR%\SAMPLES') ENDOPT (*DOC) +
AUTOFIT (*YES)
```

LNCXFER - Commande CL

La commande **LNCXFER** permet de transférer des données d'un fichier ou d'une requête de l'AS/400 vers un fichier PC au format ASCII , ou Unicode.

Paramètres

| | | |
|---------------------------------|------------------|---------|
| Fichier destination sur PC. | PCFILE | |
| Dossier / répertoire du fichier | FLR | *NONE |
| Type Source de données | CPYSRC | *FILE |
| Fichier d'origine | FROMFILE | |
| Bibliothèque | | *LIBL |
| Membre d'origine | FROMMBR | *FIRST |
| Colonnes du fichier | | *ALL |
| Conversion de types de données | | *NONE |
| Nombre maximum d'enreg. | MAXRCD | *NOMAX |
| Générer résultat en UNICODE | UNICOD | *NONE |
| Générer entêtes de colonnes | COLHDG | *NAME |
| Séparateur décimal | DECPT | *LOCALE |
| Générer fic description (lfd). | DESCF | *NO |
| Format du fichier destination | OUTFMT | *CSV |
| Caract. séparateur de colonnes | DELIMITER | *DFT |
| Encadrer entre guillemets | QUOTETXT | *YES |
| Nom du serveur LAUNCHER | EXESRV | *DFT |
| Fermer la communication | CLOSECOM | *YES |

Exemple

```
LNCXFER PCFILE (TRANFERT.TXT) FLR('c:\temp\') +
FROMFILE (LAUNCHER/SP_CUST) EXESRV (*DEV)
```

LNCFTP - Commande CL

Une extension de licence « FTP » est obligatoire.

La commande LNCFTP» permet d'adresser un serveur FTP à partir d'un programme client CL, côté AS400.

Il fait appel aux différentes fonctions de la DLL LNCFTP.

La commande réalise les opérations suivantes :

- ouverture connexion avec le serveur Launcher
- gestion de la connexion SSL ou non
- envoie des commandes au serveur FTP
- récupération des données du serveur FTP
- fermeture connexion avec le serveur Launcher

Une application sur IBM i, AS/400, commande la réception/l'envoi de fichiers de/vers un serveur FTP.

Ou bien elle envoie des commandes au serveur FTP de manière sécurisée.

Pour cela, le protocole FTP Secure (FTP over TLS/SSL) est utilisé.

L'utilisation du SSL permet d'exiger l'utilisation d'un certificat pour sécuriser la communication entre le client et le serveur.

Paramètres

| | |
|----------------------------|------------------|
| Serveur FTP | RMTSYS |
| Port | PORT |
| Connexion sécurisée | SECCNN |
| Utilisateur FTP | USER |
| Mot de passe FTP | PWD |
| Commande FTP | CMDFTP |
| Chemin distant | SRVPATH |
| Chemin local | LOCALPATH |
| Nom du PC serveur LAUNCHER | EXESRV |
| Option de fin | ENDOPT |

Exemple

```
LNCFTP      RMTSYS('milan01') PORT(990) SECCNN(*SSL) +  
            USER(aura) PWD(aura) CMDFTP(*MKDIR) +  
            SRVPATH(CED) EXESRV('192.168.15.84') +  
            ENDOPT(*NONE)
```

Les classes Java

La classe DataSource

| Attributs | Description |
|----------------------------------|--|
| String Type | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • csv: la source de données est un fichier csv. Le path complet du fichier csv est inscrit dans l'attribut Path. • oracle: la source de données est une DB Oracle • mysql: la source de données est une DB MySQL • postgresql: la source de données est une DB PostgreSQL • sybase: la source de données est une DB Sybase • as400db2: la source de données est une DB IBM DB2 pour iSeries • sqlserver: la source de données est une DB Microsoft SQL Server • saptable: la source de données provient d'une table du système SAP • sabbapi: la source de données provient d'une table résultante de l'exécution d'un module de fonction ou d'une BAPI |
| String SrvAddr | L'adresse IPv4 du serveur hébergeant la base de données Oracle, MySQL ou PostgreSQL. <u>Exemple:</u> 192.168.1.7 |
| String Port | Le port du serveur hébergeant la base de données Oracle, MySQL ou PostgreSQL. <u>Exemple:</u> 1521 pour Oracle 3306 pour MySQL 6078 pour PostgreSQL |
| String User | Le nom de l'utilisateur de la base de données. |
| String Password | Le mot de passe appartenant à l'utilisateur de la base de données. |
| String DBName | Le nom de la base de données. <u>Exemple:</u> XE pour Oracle test pour MySQL newold pour PostgreSQL |
| String Query | La requête SQL. <u>Exemple:</u> select * from test.marci where title > 'g' |
| String Path | Le path complet du fichier CSV dans le cas où Type = csv. <u>Exemple:</u> c:\\temp\\source.csv |
| String ClientSAP | Client SAP. <u>Exemple:</u> 001 |
| String LangSAP | Langue du système SAP. |
| String AshostSAP | Nom du serveur hébergeant le système SAP. <u>Exemple:</u> |

| | |
|---|---|
| | 192.168.1.7 |
| String <code>GwhostSAP</code> | Nom de la passerelle SAP (Gateway). <u>Exemple:</u> 192.168.1.7 |
| String <code>GwservSAP</code> | Numéro du port de la passerelle (Gateway). <u>Exemple:</u> 3300 |
| String <code>SysnrSAP</code> | Numéro de l'instance SAP du serveur. <u>Exemple:</u> 00 |
| String <code>TableNameSAP</code> | Nom de la table du système SAP. <u>Exemple:</u> SBOOK |
| String <code>RowskipsSAP</code> | Nombre d'enregistrements issus de la sélection dans la table SAP, dont on ne tient pas compte. <u>Exemple:</u> Si 50 enregistrements sont retournés de la recherche dans la table SAP, et que que <code>RowskipsSAP</code> = 10, alors le fichier résultat CSV ne contiendra que 40 lignes. Les 10 premiers enregistrements ne sont pas pris en compte. |
| String <code>RowcountSAP</code> | Nombre maximum d'enregistrements potentiels retournés d'une sélection dans une table SAP. <u>Exemple 1:</u> Si <code>RowcountSAP</code> = 100, et que 200 enregistrements correspondent à la sélection, alors le fichier CSV résultat ne contiendra que 100 lignes. <u>Exemple 2:</u> Si <code>RowcountSAP</code> = 100, et que 70 enregistrements correspondent à la sélection, alors le fichier CSV contiendra 70 lignes. |
| String <code>FoiSAP</code> | Champs de la table sélectionnés dans la requête. Correspond au SELECT d'une requête SQL. <u>Exemple:</u> CITYFROM,CITYTO,DISTANCE,ARRTIME,DEPTIME |
| String <code>QuerySAP</code> | Condition de sélection de la requête. Correspond au WHERE d'une requête SQL. La valeur à comparer devra être entre simple côtes. <u>Exemple:</u> CITYFROM = 'ROME' |
| String <code>BAPIName</code> | Nom de la BAPI (Business API(Application Program Interface)) ou du module de fonctions «RFC enabled». Si <code>BAPIName</code> est renseigné, les champs suivants ne sont pas pris en compte lors de l'exécution de la BAPI : <code>TableNameSAP</code> , <code>RowskipsSAP</code> , <code>RowcountSAP</code> , <code>FoiSAP</code> , <code>QuerySAP</code> . <u>Exemple:</u> BAPI_FLBOOKING_GETLIST |
| String <code>BAPIResultTable</code> | Le résultat de l'exécution de la BAPI se trouve dans différentes tables qui lui sont associées. Si <code>BAPIName</code> est renseigné, il faut également spécifié <code>BAPIResultTable</code> . <u>Exemple:</u> Pour la BAPI «BAPI_FLBOOKING_GETLIST», l'une des tables de résultat est: <code>BOOKING_LIST</code> . |
| Map<String, String> <code>ParameterMap</code> | Ce champs est optionnel. Il s'agit d'une collection de type Map qui fonctionne avec un couple clé - valeur. La clé correspond au nom du paramètre de la BAPI. La valeur est celle du paramètre de la BAPI |

| | |
|--|---|
| | <p><u>Exemple:</u> Pour la BAPI « BAPI_FLBOOKING_GETLIST », l'un des paramètres est : MAX_ROWS. Dans le code Java, on crée une collection hm dans laquelle on entre tous les paramètres. Ensuite on affecte cette collection hm au paramètre ParameterMap:</p> <pre>Map<String,String> hm = new HashMap<String,String>(); hm.put("MAX_ROWS", "70"); myDataSource.setParameterMap(hm);</pre> |
|--|---|

Tous les attributs sont accessibles et modifiables par le biais de méthodes Getters et Setters .

| Attributs | Getter (lecture) | Setter (écriture) |
|-------------------------------------|----------------------|--------------------------------------|
| String Type | getType() | setType(String) |
| String SrvAddr | getSrvAddr() | setSrvAddr(String) |
| String Port | getPort() | setPort(String) |
| String User | getUser() | setUser(String) |
| String Password | getPassword() | setPassword(String) |
| String DBName | getDBName() | setDBName(String) |
| String Query | getQuery() | setQuery(String) |
| String Path | getPath() | setPath(String) |
| String ClientSAP | getClientSAP() | setClientSAP(String) |
| String LangSAP | getLangSAP() | setLangSAP(String) |
| String AshostSAP | getAshostSAP() | setAshostSAP(String) |
| String SysnrSAP | getSysnrSAP() | setSysnrSAP(String) |
| String TableNameSAP | getTableNameSAP() | setTableNameSAP(String) |
| String RowskipsSAP | getRowskipsSAP() | setRowskipsSAP(String) |
| String RowcountSAP | getRowcountSAP() | setRowcountSAP(String) |
| String FoiSAP | getFoiSAP() | setFoiSAP(String) |
| String QuerySAP | getQuerySAP() | setQuerySAP(String) |
| String BAPIName | getBAPIName() | setBAPIName(String) |
| String BAPIResultTable | getBAPIResultTable() | setBAPIResultTable(String) |
| Map<String, String> ParameterMap | getParameterMap() | setParameterMap(Map<String, String>) |

La classe LNCPRDOC

| Attributs | Description |
|---------------------------------------|---|
| String LNCSrvAddr | Adresse IPv4 du serveur Launcher. <u>Exemple:</u> 192.168.1.3 |
| String Doc | Nom du document modèle. <u>Exemple:</u> C:\\temp\\template.docx |
| String WordPath | Répertoire par défaut de Word. |
| String MrgType | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NONE : aucune fusion n'est faite. • *FILE : les données à fusionner sont issues d'un fichier csv ou d'une base de données. |
| String Unlink | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : suppression des champs vides • False : aucune suppression |
| String OutSave | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : sauvegarde du document généré • False : aucune sauvegarde |
| String MrgSel | Format de fusion : <ul style="list-style-type: none"> • *LTR : format lettre • *CAT : format catalogue |
| String OutPdf | Chemin complet du fichier PDF à sauvegarder. |
| String OutPrn | Chemin complet du fichier PRN à sauvegarder. |
| String ExecMrg | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : exécution du publipostage • False : le document est simplement préparé pour le publipostage |
| String Printer | Nom de l'imprimante. |
| String Copies | Nombre de copies générées. |
| String Tray | Nom du bac. |
| String FirstPageTray | Numéro ou nom de bac. |
| String OtherPageTray | Numéro ou nom de bac. |
| String DirectPrt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : impression directe • False : pas d'impression directe |
| String OneDoc | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : merge document temporaire • False : aucun merge |
| String DltLgn | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : suppression des lignes vides après la fusion • False : aucune suppression |
| String MailTo | Un champs ou rubrique d'un fichier csv ou d'une table d'une base de données contenant des adresses email. |
| String MailSubj | Sujet du mail. |
| String HeaderSrc | Nom du fichier qui contient les entêtes. |
| String TmpPdf | Fichier temporaire PDF. |
| String TmpPrn | Fichier temporaire PRN. |
| String DataSource | Le chemin complet du fichier CSV si on n'utilise pas la classe DataSource. |
| DataSource sourceData | Une instance de la classe DataSource. |

| | |
|-----------------------|---|
| String Output | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *PDF : si le document généré est au format PDF • *EMAIL : si le résultat est envoyé par email • *OUTPRN : fichier d'impression PRN • *PRINT : le document sera imprimé |
| String ShowDoc | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : le document généré est visible • False : le document généré n'est pas visible |
| String SavDoc | Chemin complet du fichier résultat. Exemple: C:\\temp\\result.docx |
| String SavFmt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NORMAL : format du fichier modèle • *DOC : format Word • *HTM : format HTML |
| String EndOpt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NONE : aucune option de fin de traitement • *DOC : fermeture du fichier Word • *ALL : fin de la session de commandes et fermeture de l'application Word • *APP : fermeture de l'application Word |

Tous les attributs sont accessibles et modifiables par le biais de méthodes Getters et Setters .

| Attributs | Getter (lecture) | Setter (écriture) |
|-----------------------------|--------------------|--------------------|
| String LNCSrvAddr | getLNCSrvAddr() | setLNCSrvAddr() |
| String Doc | getDoc() | setDoc() |
| String WordPath | getWordPath() | setWordPath() |
| String MrgType | getMrgType() | setMrgType() |
| String Unlink | getUnlink() | setUnlink() |
| String OutSave | getOutsave() | setOutSave() |
| String MrgSel | getMrgSel() | setMrgSel() |
| String OutPdf | getOutPdf() | setOutPdf() |
| String OutPrn | getOutPrn() | setOutPrn() |
| String ExecMrg | getExecMrg() | setExecMrg() |
| String Printer | getPrinter() | setPrinter() |
| String Copies | getCopies() | setCopies() |
| String Tray | getTray() | setTray() |
| String FirstPageTray | getFirstPageTray() | setFirstPageTray() |
| String OtherPageTray | getOtherPageTray() | setOtherPageTray() |
| String DirectPrt | getDirectPrt() | setDirectPrt() |
| String OneDoc | getOneDoc() | setOneDoc() |
| String DltLgn | getDltLgn() | setDltLgn() |
| String MailTo | getMailTo() | setMailTo() |
| String MailSubj | getMailSubj() | setMailSubj() |
| String HeaderSrc | getHeaderSrc() | setHeaderSrc() |
| String TmpPdf | getTmpPdf() | setTmpPdf() |
| String TmpPrn | getTmpPrn() | setTmpPrn() |
| String DataSource | getDataSource() | setDataSource() |

| | | |
|------------------------------------|------------------------------|------------------------------|
| String <code>Output</code> | <code>getOutput()</code> | <code>setOutput()</code> |
| String <code>ShowDoc</code> | <code>getShowDoc()</code> | <code>setShowDoc()</code> |
| String <code>SavDoc</code> | <code>getSavDoc()</code> | <code>setSavDoc()</code> |
| String <code>SavFmt</code> | <code>getSavFmt()</code> | <code>setSavFmt()</code> |
| String <code>EndOpt</code> | <code>getEndOpt()</code> | <code>setEndOpt()</code> |
| DataSource <code>sourceData</code> | <code>getSourceData()</code> | <code>setSourceData()</code> |

La methode **execute()** permet de procéder à l'exécution de la commande LNCPRDOC.

La classe LNCTOXLS

| Attributs | Description |
|-----------------------------------|---|
| String LNCSrvAddr | Adresse IPv4 du serveur Launcher. <u>Exemple:</u> 192.168.1.3 |
| String ToXls | Nom du document modèle. <u>Exemple:</u> C:\\temp\\template.xlsx |
| String ToSheet | Nom de la feuille Excel. |
| String ToName | Nom de la plage de destination. |
| String XlsOpt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *REPLACE : les nouvelles données remplacent les anciennes données. • *RESIZE : les nouvelles données remplacent les anciennes données. La zone dans le classeur est redimensionnée. • *ADD : les nouvelles données sont ajoutées aux anciennes données. • *INSERT : des lignes sont insérées pour recevoir les nouvelles données. Le paramètre ENTIRER détermine si des lignes entières sont insérées, ou non. • *MAP : les valeurs des colonnes de la source de données sont affectées aux cellules portant les mêmes noms dans le tableau Excel. Seul le premier entregistrement de la source de donnée est utilisé. |
| String NoData | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *BLANK : si aucune donnée n'est présente dans la source de données, la zone est laissée vide • *DEL : si aucune donnée n'est présente dans la source de données, les cellules sont supprimées |
| String XlsMap | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *MAPNAME : les valeurs des colonnes de la source de données sont affectées aux cellules portant les mêmes noms dans le tableau Excel. • *MAPVALUE : les valeurs des colonnes de la source de données sont affectées aux cellules dont la valeur initiale est identique |
| String ColPref | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NONE : aucun préfixe pour les colonnes • Valeur du préfixe pour les colonnes |
| String DataSource | Le path complet du fichier CSV si on n'utilise pas la classe DataSource. |
| String RecCnt | Si on utilise directement un fichier CSV comme source de données, il faut obligatoirement spécifier le nombre de lignes du fichier CSV. Dans tous les autres cas, cet attribut n'est pas utilisé. |
| String Entirer | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : des lignes entières sont insérées • False : pas d'insertion de lignes entières |

| | |
|---------------------------------------|--|
| String AutoFmt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NONE : aucun format automatique n'est appliqué. • nom du format : le format spécifié est appliqué. |
| String AutoFit | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : ajuster automatiquement la largeur des colonnes. • False : ne pas modifier la largeur des colonnes. |
| String FmtCells | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : le format des cellules est fixé en fonction du type Base de Données. • False : le format des cellules est fixé par Excel, en fonction des valeurs. |
| String AddColH | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : les noms des colonnes sont insérés en première ligne. • False : les noms des colonnes ne sont pas insérés en première ligne. |
| String Freeze | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : remplace les expressions par leur valeur calculée • False : aucun remplacement |
| String LocateSheet | Spécifie la feuille active. |
| String LocateCell | Spécifie la cellule active. |
| DataSource sourceData | Une instance de la classe DataSource. |
| String Output | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *PRINT : le document sera imprimé sur l'imprimante définie par défaut. |
| String ShowDoc | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : le document généré est visible • False : le document généré n'est pas visible |
| String SavDoc | Chemin complet du fichier résultat. <u>Exemple:</u> C:\\temp\\result.xlsx |
| String SavFmt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NORMAL : format du fichier modèle • *XLS : format Excel • *TXT : format texte |
| String EndOpt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NONE : aucune option de fin de traitement • *DOC : fermeture du fichier Excel • *ALL : fin de la session de commandes et fermeture de l'application Excel • *APP : fermeture de l'application Excel |

Tous les attributs sont accessibles et modifiables par le biais de méthodes Getters et Setters .

| Attributs | Getter (lecture) | Setter (écriture) |
|---------------------------------------|------------------|-------------------|
| String LNCSrvAddr | getLNCSrvAddr() | setLNCSrvAddr() |
| String ToXls | getToXls() | setToXls() |
| String ToSheet | getToSheet() | setToSheet() |
| String ToName | getToName() | setToName() |
| String XlsOpt | getXlsOpt() | setXlsOpt() |
| String NoData | getNoData() | setNoData() |
| String XlsMap | getXlsMap() | setXlsMap() |
| String ColPref | getColPref() | setColPref() |
| String DataSource | getDataSource() | setDataSource() |
| String RecCnt | getRecCnt() | setRecCnt() |
| String Entirer | getEntirer() | setEntirer() |
| String AutoFmt | getAutoFmt() | setAutoFmt() |
| String AutoFit | getAutoFit() | setAutoFit() |
| String FmtCells | getFmtCells() | setFmtCells() |
| String AddColH | getAddColH() | setAddColH() |
| String Freeze | getFreeze() | setFreeze() |
| String LocateSheet | getLocateSheet() | setLocateSheet() |
| String LocateCell | getLocateCell() | setLocateCell() |
| DataSource sourceData | getSourceData() | setSourceData() |
| String Output | getOutput() | setOutput() |
| String ShowDoc | getShowDoc() | setShowDoc() |
| String SavDoc | getSavDoc() | setSavDoc() |
| String SavFmt | getSavFmt() | setSavFmt() |
| String EndOpt | getEndOpt() | setEndOpt() |

La methode **execute()** permet de procéder à l'exécution de la commande LNCTOXLS.

Exemples d'utilisation

Ces exemples concernent les applications Java exécutées sur le serveur Launcher.

L'adresse du serveur (attribut LNCSrvAddr) Launcher est : **192.168.1.9**.

Le code Java devra obligatoirement contenir les lignes suivantes:

```
import fr.aura.launcher.wrapper.DataSource ;
import fr.aura.launcher.wrapper.LNCTOXLS ;
import fr.aura.launcher.wrapper.LNCPRTDOC ;
```

1) LNCTOXLS : fichier CSV comme source de données

Dans cet exemple, la source de données est un fichier CSV.

La commande **LNCTOXLS** est utilisée pour générer un document Excel .

```
String Destination="c:\\temp\\destination.xls";

DataSource myDataSource = new DataSource();
myDataSource.setType("csv");
myDataSource.setPath("c:\\temp\\source_csv");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setToSheet("*NONE");
myLNCTOXLS.setToName("*NONE");
myLNCTOXLS.setColPref("*NONE");
myLNCTOXLS.setAutoFmt("*NONE");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*XLS");
myLNCTOXLS.setSavDoc(Destination);
myLNCTOXLS.setRecCnt(55); //Obligatoire si Type = csv

myLNCTOXLS.execute();
```

2) LNCTOXLS : Oracle

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **Oracle**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 1521
- Utilisateur DB : hr
- Mot de passe utilisateur DB : hr
- Nom de la database : XE
- Requête SQL : select * from countries

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\lnctoxls_test2.xls

```
String Destination="C:\\temp\\lnctoxls_test2.xls";

DataSource myDataSource = new DataSource();
myDataSource.setType("oracle");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("1521");
myDataSource.setUser("hr");
myDataSource.setPassword("hr");
myDataSource.setDBName("XE");
myDataSource.setQuery("select * from countries");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setToSheet("*NONE");
myLNCTOXLS.setToName("*NONE");
myLNCTOXLS.setColPref("*NONE");
myLNCTOXLS.setAutoFmt("*NONE");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*XLS");
myLNCTOXLS.setSavDoc(Destination);

myLNCTOXLS.execute();
```

3) LNCTOXLS : MySQL

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **MySQL**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 3306
- Utilisateur DB : root
- Mot de passe utilisateur DB : (aucun)
- Nom de la database : test
- Requête SQL : select * from test.marci

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\lnctoxls_test3.xls

```
String Destination="C:\\temp\\lnctoxls_test3.xls";

DataSource myDataSource = new DataSource();
myDataSource.setType("mysql");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("3306");
myDataSource.setUser("root");
myDataSource.setPassword("");
myDataSource.setDBName("test");
myDataSource.setQuery("select * from test.marci");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setToSheet("*NONE");
myLNCTOXLS.setToName("*NONE");
myLNCTOXLS.setColPref("*NONE");
myLNCTOXLS.setAutoFmt("*NONE");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*XLS");
myLNCTOXLS.setSavDoc(Destination);

myLNCTOXLS.execute();
```

4) LNCTOXLS : PostgreSQL

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **PostgreSQL**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 6078
- Utilisateur DB : postgres
- Mot de passe utilisateur DB : aura
- Nom de la database : newold
- Requête SQL : select * from merci where title < 'b'

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\lnctoxls_test4.xls

```
String Destination="C:\\temp\\lnctoxls_test4.xls";

DataSource myDataSource = new DataSource();
myDataSource.setType("postgresql");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("6078");
myDataSource.setUser("postgres");
myDataSource.setPassword("aura");
myDataSource.setDBName("newold");
myDataSource.setQuery("select * from merci where title < 'b'");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setToSheet("*NONE");
myLNCTOXLS.setToName("*NONE");
myLNCTOXLS.setColPref("*NONE");
myLNCTOXLS.setAutoFmt("*NONE");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*XLS");
myLNCTOXLS.setSavDoc(Destination);

myLNCTOXLS.execute();
```

5) LNCTOXLS : Sybase

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **Sybase**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 2638
- Utilisateur DB : dba
- Mot de passe utilisateur DB : sql
- Nom de la database : newold
- Requête SQL : SELECT ID, GivenName, Surname FROM Customers

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\lnctoxls_test5.xls

```
String Destination="C:\\temp\\lnctoxls_test5.xls";

DataSource myDataSource = new DataSource();
myDataSource.setType("sybase");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("2638");
myDataSource.setUser("dba");
myDataSource.setPassword("sql");
myDataSource.setDBName("newold");
myDataSource.setQuery("SELECT ID, GivenName, Surname FROM
Customers ");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setToSheet("*NONE");
myLNCTOXLS.setToName("*NONE");
myLNCTOXLS.setColPref("*NONE");
myLNCTOXLS.setAutoFmt("*NONE");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*XLS");
myLNCTOXLS.setSavDoc(Destination);

myLNCTOXLS.execute();
```

6) LNCTOXLS : DB2 for iSeries

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **DB2 pour iSeries**
- Adresse de l'AS400 : 192.168.1.7
- Port du serveur DB : pas nécessaire
- Utilisateur DB : qpgmr
- Mot de passe utilisateur DB : aura
- Nom des bibliothèques : easycomxmp
- Requête SQL : select * from easycomxmp.s_customer

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\lnctoxls_test4.xls

```
String Destination="C:\\temp\\lnctoxls_test4.xls";

DataSource myDataSource = new DataSource();
myDataSource.setType("as400db2");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setUser("qpgmr");
myDataSource.setPassword("aura");
myDataSource.setDBName("easycomxmp");
myDataSource.setQuery("select * from easycomxmp.s_customer");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setToSheet("*NONE");
myLNCTOXLS.setToName("*NONE");
myLNCTOXLS.setColPref("*NONE");
myLNCTOXLS.setAutoFmt("*NONE");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*XLS");
myLNCTOXLS.setSavDoc(Destination);

myLNCTOXLS.execute();
```

7) LNCTOXLS : Microsoft SQL Server

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **Microsoft SQL Server**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 1433
- Utilisateur DB : sa
- Mot de passe utilisateur DB : aura
- Nom de la database : master
- Requête SQL : select * from dbo.spt_values

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\lnctoxls_test4.xls

```
String Destination="C:\\temp\\lnctoxls_test4.xls";

DataSource myDataSource = new DataSource();
myDataSource.setType("sqlserver");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("1433");
myDataSource.setUser("sa");
myDataSource.setPassword("aura");
myDataSource.setDBName("master");
myDataSource.setQuery("select * from dbo.spt_values");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setToSheet("*NONE");
myLNCTOXLS.setToName("*NONE");
myLNCTOXLS.setColPref("*NONE");
myLNCTOXLS.setAutoFmt("*NONE");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*XLS");
myLNCTOXLS.setSavDoc(Destination);

myLNCTOXLS.execute();
```

8) LNCTOXLS : SAP

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB: **SAP**
- Adresse du serveur SAP : 192.168.1.7
- Utilisateur SAP: DDIC
- Mot de passe utilisateur SAP: aura
- Nom de la table SAP: SPFLI
- Requête SQL: select **CITYFROM,CITYTO,DISTANCE,DEPTIME** from **SBOOK** where **CITYFROM = 'ROME'**

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\sap_test.xls

```
String Destination="C:\\temp\\sap_test.xlsx";

DataSource myDataSource = new DataSource();
myDataSource.setType("saptable");
myDataSource.setUser("DDIC");
myDataSource.setPassword("aura");
myDataSource.setAshostSAP("192.168.1.7");
myDataSource.setClientSAP("001");
myDataSource.setLangSAP("");
myDataSource.setSysnrSAP("00");
myDataSource.setGwhostSAP("192.168.1.7");
myDataSource.setGwservSAP("3300");
myDataSource.setQuerySAP("CITYFROM = 'ROME'");
myDataSource.setRowcountSAP("5");
myDataSource.setRowskipsSAP("0");
myDataSource.setFoisSAP("CITYFROM, CITYTO, DISTANCE, DEPTIME");
myDataSource.setTableNameSAP("SPFLI");

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls("*new");
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setFmtCells("false");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("true");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*NORMAL");
myLNCTOXLS.setSavDoc(Destination);

myLNCTOXLS.execute();
```

9) LNCTOXLS : SAP – Utilisation d'une BAPI

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB: **SAP**
- Adresse du serveur SAP : 192.168.1.7
- Utilisateur SAP: DDIC
- Mot de passe utilisateur SAP: aura
- Nom de la BAPI: **BAPI_FLBOOKING_GETLIST**
- Table résultat utilisée pour exporter vers Excel: BOOKING_LIST
- Paramètre de la BAPI utilisé: MAX_ROWS=7

La commande **LNCTOXLS** est utilisée pour générer le document Excel suivant :
C:\temp\sap_test.xlsx

```
String Destination="C:\\temp\\sap_test.xlsx";
String template="C:\\temp\\SBOOK_template_BAPI.xlsx";

// Création d'une liste pour les paramètres de la BAPI
Map<String,String> hm = new HashMap<String,String>();

// On met les paramètres dans la liste
hm.put("MAX_ROWS", "70");

DataSource myDataSource = new DataSource();
myDataSource.setType("sapbapi");
myDataSource.setUser("DDIC");
myDataSource.setPassword("aura");
myDataSource.setAshostSAP("192.168.1.7");
myDataSource.setClientSAP("001");
myDataSource.setLangSAP("");
myDataSource.setSysnrSAP("00");
myDataSource.setGwhostSAP("192.168.1.7");
myDataSource.setGwservSAP("3300");
myDataSource.setBAPIName("BAPI_FLBOOKING_GETLIST");
myDataSource.setBAPIResultTable("BOOKING_LIST");
myDataSource.setParameterMap(hm);

LNCTOXLS myLNCTOXLS = new LNCTOXLS();
myLNCTOXLS.setToXls(template);
myLNCTOXLS.setAutoFit("true");
myLNCTOXLS.setAddColH("false");
myLNCTOXLS.setShowDoc("false");
myLNCTOXLS.setSourceData(myDataSource);
myLNCTOXLS.setLNCSrvAddr("192.168.1.9");
myLNCTOXLS.setSavFmt("*NORMAL");
```

```
myLNCTOXLS.setSavDoc(Destination);  
myLNCTOXLS.setToName("Tableau1");  
myLNCTOXLS.setToSheet("Invoice");  
myLNCTOXLS.setXlsOpt("*resize");  
myLNCTOXLS.setXlsMap("*MAPNAME");
```

```
myLNCTOXLS.execute();
```

10) LNCPRTRDOC : fichier CSV comme source de données

Dans cet exemple, la source de données est un fichier CSV (C:\temp\LNC002tmp.csv).

La commande **LNCPRTRDOC** est utilisée pour générer le document Word :

C:\temp\TEST1_csv.DOC

Le modèle utilisé pour le publipostage est le suivant :

C:\temp\SP_CUST.DOC

```
String source = "C:\\temp\\LNC002tmp.csv";
String template = "C:\\temp\\SP_CUST.DOC";
String destination = "C:\\temp\\TEST1_csv.DOC";

DataSource myDataSource = new DataSource();
myDataSource.setType("csv");
myDataSource.setPath(source);

LNCPRTRDOC myLNCPRTRDOC = new LNCPRTRDOC();
myLNCPRTRDOC.setDoc(template);
myLNCPRTRDOC.setSavFmt("*NORMAL");
myLNCPRTRDOC.setShowDoc("true");
myLNCPRTRDOC.setMrgType("*FILE");
myLNCPRTRDOC.setUnlink("true");
myLNCPRTRDOC.setOutSave("true");
myLNCPRTRDOC.setSavDoc(destination);
myLNCPRTRDOC.setMrgSel("*LTR");
myLNCPRTRDOC.setExecMrg("true");
myLNCPRTRDOC.setEndOpt("*CON");
myLNCPRTRDOC.setHeaderSrc("*FILE");
myLNCPRTRDOC.setSourceData(myDataSource);
myLNCPRTRDOC.setLNCSrvAddr("192.168.1.9");

myLNCPRTRDOC.execute();
```

11) LNCPRDOC : Oracle

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **Oracle**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 1521
- Utilisateur DB : hr
- Mot de passe utilisateur DB : hr
- Nom de la database : XE
- Requête SQL : select * from employees

La commande **LNCPRDOC** est utilisée pour générer le document Word suivant :

C:\temp\TEST2_oracle.doc

Le modèle utilisé pour le publipostage est le suivant :

C:\temp\SP_CUST_ORACLE.DOC

```
String template = "C:\\temp\\SP_CUST_ORACLE.DOC";
String destination = "C:\\temp\\TEST2_oracle.DOC";

DataSource myDataSource = new DataSource();
myDataSource.setType("oracle");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("1521");
myDataSource.setUser("hr");
myDataSource.setPassword("hr");
myDataSource.setDBName("XE");
myDataSource.setQuery("select * from employees");

LNCPRDOC myLNCPRDOC = new LNCPRDOC();
myLNCPRDOC.setDoc(template);
myLNCPRDOC.setSavFmt("*NORMAL");
myLNCPRDOC.setShowDoc("true");
myLNCPRDOC.setMrgType("*FILE");
myLNCPRDOC.setOutSave("true");
myLNCPRDOC.setSavDoc(destination);
myLNCPRDOC.setMrgSel("*LTR");
myLNCPRDOC.setExecMrg("true");
myLNCPRDOC.setHeaderSrc("*FILE");
myLNCPRDOC.setSourceData(myDataSource);
myLNCPRDOC.setDltLgn("true");
myLNCPRDOC.setSourceData(myDataSource);
myLNCPRDOC.setLNCSrvAddr("192.168.1.9");

myLNCPRDOC.execute();
```

12) LNCPRDOC : MySQL

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **MySQL**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 3306
- Utilisateur DB : root
- Mot de passe utilisateur DB : (aucun)
- Nom de la database : test
- Requête SQL : select * from test.marci

La commande **LNCPRDOC** est utilisée pour générer le document Word suivant :

C:\temp\TEST2_MySQL.doc

Le modèle utilisé pour le publipostage est le suivant :

C:\temp\SP_CUST_MYSQL.DOC

```
String template = "C:\\temp\\SP_CUST_MYSQL.DOC";
String destination = "C:\\temp\\TEST2_MySQL.DOC";

DataSource myDataSource = new DataSource();
myDataSource.setType("mysql");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("3306");
myDataSource.setUser("root");
myDataSource.setPassword("");
myDataSource.setDBName("test");
myDataSource.setQuery("select * from test.marci");

LNCPRDOC myLNCPRDOC = new LNCPRDOC();
myLNCPRDOC.setDoc(template);
myLNCPRDOC.setSavFmt("*NORMAL");
myLNCPRDOC.setShowDoc("true");
myLNCPRDOC.setMrgType("*FILE");
myLNCPRDOC.setOutSave("true");
myLNCPRDOC.setSavDoc(destination);
myLNCPRDOC.setMrgSel("*LTR");
myLNCPRDOC.setExecMrg("true");
myLNCPRDOC.setHeaderSrc("*FILE");
myLNCPRDOC.setSourceData(myDataSource);
myLNCPRDOC.setSourceData(myDataSource);
myLNCPRDOC.setLNCSrvAddr("192.168.1.9");

myLNCPRDOC.execute();
```



13) LNCPRDOC : PostgreSQL

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB : **PostgreSQL**
- Adresse du serveur DB : 192.168.1.7
- Port du serveur DB : 6078
- Utilisateur DB : postgres
- Mot de passe utilisateur DB : aura
- Nom de la database : newold
- Requête SQL : select * from merci where title < 'b'

La commande **LNCPRDOC** est utilisée pour générer le document Word suivant :

C:\temp\TEST4_POSTGRESQL.doc

Le modèle utilisé pour le publipostage est le suivant :

C:\temp\SP_CUST_POSTGRESQL.DOC

```
String template = "C:\\temp\\SP_CUST_POSTGRESQL.DOC";
String destination = "C:\\temp\\TEST4_POSTGRESQL.DOC";

DataSource myDataSource = new DataSource();
myDataSource.setType("postgresql");
myDataSource.setSrvAddr("192.168.1.7");
myDataSource.setPort("6078");
myDataSource.setUser("postgres");
myDataSource.setPassword("aura");
myDataSource.setDBName("newold");
myDataSource.setQuery("select * from merci where title < 'b'");

LNCPRDOC myLNCPRDOC = new LNCPRDOC();
myLNCPRDOC.setDoc(template);
myLNCPRDOC.setSavFmt("*NORMAL");
myLNCPRDOC.setShowDoc("true");
myLNCPRDOC.setMrgType("*FILE");
myLNCPRDOC.setOutSave("true");
myLNCPRDOC.setSavDoc(destination);
myLNCPRDOC.setMrgSel("*LTR");
myLNCPRDOC.setExecMrg("true");
myLNCPRDOC.setEndOpt("*NONE");
myLNCPRDOC.setHeaderSrc("*FILE");
myLNCPRDOC.setSourceData(myDataSource);
myLNCPRDOC.setSourceData(myDataSource);
myLNCPRDOC.setLNCSrvAddr("192.168.1.9");

myLNCPRDOC.execute();
```

14) LNCPRDOC : SAP - Utilisation d'une BAPI

Dans cet exemple, la base de donnée adressée a les caractéristiques suivantes :

- Type DB: **SAP**
- Adresse du serveur SAP : 192.168.1.7
- Utilisateur SAP: DDIC
- Mot de passe utilisateur SAP: aura
- Nom de la BAPI: **BAPI_FLBOOKING_GETLIST**
- Table résultat utilisée pour exporter vers Excel: BOOKING_LIST
- Paramètre de la BAPI utilisé: MAX_ROWS=7

La commande **LNCPRDOC** est utilisée pour générer le document Word suivant :

C:\temp\result.docx

Le modèle utilisé pour le publipostage est le suivant :

C:\temp\template.docx

```
String template = "C:\\temp\\template.docx";
String destination = "C:\\temp\\result.docx";

// Création d'une liste pour les paramètres de la BAPI
Map<String,String> hm = new HashMap<String,String>();

// On met les paramètres dans la liste
hm.put("MAX_ROWS", "70");

DataSource myDataSource = new DataSource();
myDataSource.setType("sapbapi");
myDataSource.setUser("DDIC");
myDataSource.setPassword("aura");
myDataSource.setAshostSAP("192.168.1.7");
myDataSource.setClientSAP("001");
myDataSource.setLangSAP("");
myDataSource.setSysnrSAP("00");
myDataSource.setGwhostSAP("192.168.1.7");
myDataSource.setGwservSAP("3300");
myDataSource.setBAPIName("BAPI_FLBOOKING_GETLIST");
myDataSource.setBAPIResultTable("BOOKING_LIST");
myDataSource.setParameterMap(hm);

LNCPRDOC myLNCPRDOC = new LNCPRDOC();
myLNCPRDOC.setDoc(template);
myLNCPRDOC.setSavFmt("*NORMAL");
myLNCPRDOC.setShowDoc("false");
myLNCPRDOC.setMrgType("*FILE");
myLNCPRDOC.setOutSave("true");
myLNCPRDOC.setSavDoc(destination);
myLNCPRDOC.setMrgSel("*LTR");
```

```
myLNCPRTDOC.setExecMrg("true");  
myLNCPRTDOC.setEndOpt("*NONE");  
myLNCPRTDOC.setHeaderSrc("*FILE");  
myLNCPRTDOC.setSourceData(myDataSource);  
myLNCPRTDOC.setLNCSrvAddr("192.168.1.9");  
  
myLNCPRTDOC.execute();
```

Les webservices

Appel des méthodes du web service

Une fois que le web service est démarré, les méthodes du service Launcher REST Gateway sont accessibles via l'URL **https://<IP machine>:9090/launcher/nom_méthode** (le numéro de port pouvant varier selon la valeur spécifié par l'utilisateur dans beans.xml ou avec le Service Manager).

Les méthodes principales du service sont :

- open
- ldapauth
- cmd
- close
- filetransfer
- lncprtdoc
- lnctoxls

Toutes ces méthodes peuvent être appelées par un client connecté au même réseau que celui sur lequel tourne Launcher REST Gateway. La plupart de ces méthodes sont accessibles par des requêtes HTTPS **POST** ou **GET**. S'il faut passer des paramètres à la méthode, ils sont inclus dans le **corps de message POST en format JSON**. Certains paramètres tels que **l'ID de la connexion Launcher** peuvent figurer dans **l'URL du service**.

Open

La méthode **open** établit une connexion avec le serveur Launcher.

Si « Useldap=Yes » dans le fichier **launcher.ini**, cette méthode permet aussi d'authentifier (LDAP) le client auprès du web service Launcher REST Gateway. Il doit inclure son nom d'utilisateur et mot de passe (LDAP – Active Directory) dans le corps de la requête pour s'authentifier.

Voici un exemple d'un appel à la méthode **open** qui permet d'établir une connexion au serveur Launcher hébergé sur localhost au port numéro 6078 (qui sont d'ailleurs les valeurs par défaut) :

```
POST /launcher/open HTTP/1.1
Host: localhost:9090
Content-Type: application/JSON
```

```
{  "open":  {  "server": "localhost",  "port": "6078",  "username": "aura",  "password": "aura"  } }
```

La requête POST ci-dessus est envoyée à l'URL **https://<IP machine>:9090/launcher/open**.

Si l'authentification réussit et que Launcher REST Gateway parvient à établir une connexion avec le serveur Launcher, un entier représentant **l'ID du client (de la connexion Launcher)** sera retourné au client. Le client doit **inclure cet ID dans toutes les requêtes ultérieures**.



Ldapauth

La méthode `ldapauth` sert à **authentifier le client auprès du serveur Launcher, qui est configuré pour une authentification LDAP**. Cette méthode est accessible à l'URL `https://localhost:9090/launcher/ldapauth`. La requête POST faite à cet URL doit contenir un message JSON dans le corps, précisant le nom d'utilisateur Active Directory et son mot de passe. Voici un exemple :

```
POST /launcher/ldapauth HTTP/1.1
Host: localhost:9090
Content-Type: application/JSON
Cache-Control: no-cache
```

```
{ "ldapauth": { "id": "1", "username": "john", "password": "secretpassword" } }
```

Cmd

La méthode `cmd` sert à **exécuter des commandes CMD de Launcher**. La méthode `cmd` est appelée par requête POST car les paramètres de la commande, **parm1** et **parm2**, sont inclus dans le corps du message.

Pour appeler la méthode `cmd`, il faut inclure l'ID du client, la commande, `parm1` et `parm2` dans le message JSON dans le corps de la requête POST comme dans l'exemple suivant :

```
POST /launcher/cmd HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Cache-Control: no-cache
```

```
{"cmd": {"verb": "WORDOPEN", "parm1": "new", "parm2": "visible", "id": "1"}}
```

Une fois la commande exécutée, **la valeur retournée par le serveur Launcher est envoyée au client par Launcher REST Gateway** :

- *ACK ou *MSG si la commande s'est correctement déroulée.
- *REJ si la commande a rencontré un problème. Plus d'infos seront disponibles dans la trace du serveur Launcher.

Filetransfer

La méthode `filetransfer` permet de transférer des fichiers depuis la machine du client au serveur Launcher. Lors de l'appel de cette méthode, les données à transmettre ne sont pas écrites en format JSON, mais sont plutôt incluses dans le **form-data**. Les données à inclure lors d'un appel à `filetransfer` sont **l'ID du client**, le **nom de fichier de destination**, le **répertoire de destination**, et le **fichier même**. En voici un exemple :

```
POST /launcher/filetransfer HTTP/1.1
Host: localhost:9090
Cache-Control: no-cache
----WebKitFormBoundaryE19zNvXGzXaLv55C
Content-Disposition: form-data; name="id"
1
```

```
----WebKitFormBoundaryE19zNvXGzXaLvS5C  
Content-Disposition: form-data; name="destFileName"  
Example_doc.pdf
```

```
----WebKitFormBoundaryE19zNvXGzXaLvS5C  
Content-Disposition: form-data; name="destDir"  
C:\Temp
```

```
----WebKitFormBoundaryE19zNvXGzXaLvS5C  
Content-Disposition: form-data; name="file"  
C:\Temp\test.pdf
```

L'exemple ci-dessus transfèrera un fichier au serveur Launcher, où il sera sauvegardé dans le dossier C:\Temp sous le nom Example_doc.pdf. Notez que le chemin complet du fichier destination, incluant le chemin du répertoire, peut être indiqué dans le champ « destFileName » et dans ce cas le champ « destDir » peut être omis.

L'inclusion du fichier dans le form-data peut être faite de plusieurs manières selon le langage avec lequel le client a été codé. Regardez les exemples dans le répertoire d'installation.

Close

La méthode **close** sert à fermer la connexion avec le serveur Launcher. Le seul paramètre à passer lors de l'appel à cette méthode est l'ID du client. Ceci peut être fait avec une requête GET contenant l'ID du client dans l'URL comme la suivante :

```
GET /launcher/close/1 HTTP/1.1  
Host: localhost:9090  
Content-Type: application/json  
Cache-Control: no-cache
```

La requête dans cet exemple fermera la connexion du client avec l'ID 1.

Autrement, la méthode **close** peut aussi être exécutée en faisant une requête POST contenant l'ID du client dans le corps du message :

```
POST /launcher/close HTTP/1.1  
Host: localhost:9090  
Content-Type: application/json  
Cache-Control: no-cache
```

```
{ "close": { "id": "1" } }
```

Incprtdoc

La méthode **Incprtdoc** sert à **exécuter un publipostage Word à partir d'un modèle et d'une source de données (fichier csv)**. La méthode **Incprtdoc** est appelée par requête POST car les paramètres de la commande sont inclus dans le corps du message.

L'URL suivante est utilisée : [/launcher/Incprtdoc](#)



Voici les différents paramètres :

```
<command name="LNCPRDOC">
  <param1>
    <param name="Doc" default="*New"/>
    <param name="WordPath" default="notrequired"/>
    <param name="MrgType" default="*FILE"/>
    <param name="UnLink" default="true"/>
    <param name="DataSource" default="" type="string"/>
    <param name="OutSave" default="false"/>
    <param name="MrgSel" default="*LTR"/>
    <param name="SavFmt" default="*NORMAL"/>
    <param name="ExecMrg" default="true"/>
    <param name="DltLgn" default="true"/>
    <param name="ShowDoc" default="false"/>
    <param name="SavDoc" default="" type="string"/>
  </param1>
</command>
```

| Paramètres | Description |
|--------------------------|---|
| String Doc | Nom du document modèle. <u>Exemple:</u> C:\\temp\\template.docx |
| String WordPath | Répertoire par défaut de Word. |
| String MrgType | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NONE : aucune fusion n'est faite. • *FILE : les données à fusionner sont issues d'un fichier csv ou d'une base de données. |
| String Unlink | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : suppression des champs vides • False : aucune suppression |
| String OutSave | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : sauvegarde du document généré • False : aucune sauvegarde |
| String MrgSel | Format de fusion : <ul style="list-style-type: none"> • *LTR : format lettre • *CAT : format catalogue |
| String ExecMrg | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : exécution du publipostage • False : le document est simplement préparé pour le publipostage |
| String DltLgn | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : suppression des lignes vides après la fusion • False : aucune suppression |
| String DataSource | Le path complet du fichier CSV. |
| String ShowDoc | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • True : le document généré est visible • False : le document généré n'est pas visible |
| String SavDoc | Chemin complet du fichier résultat. <u>Exemple:</u> |

| | |
|-------------------------------|---|
| | C:\\temp\\result.docx |
| String SavFmt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *NORMAL : format du fichier modèle • *DOC : format Word • *HTM : format HTML |

Lnctoxls

La méthode **Lnctoxls** sert à **exporter des données (fichier csv) vers un modèle Excel**.

La méthode **Lnctoxls** est appelée par requête POST car les paramètres de la commande sont inclus dans le corps du message.

L'URL suivante est utilisée : </launcher/lnctoxls>

Voici les différents paramètres :

```

<command name="LNCTOxls">
  <param1>
    <param name="ToXls" default="*New"/>
    <param name="ToSheet" default=""/>
    <param name="ToName" default="*NONE"/>
    <param name="ColPref" default="*NONE"/>
    <param name="DataSource" default="" type="string"/>
    <param name="RecCnt" default=""/>
    <param name="AutoFmt" default="*NONE"/>
    <param name="AutoFit" default=""/>
    <param name="FmtCells" default=""/>
    <param name="AddColH" default="*NONE"/>
    <param name="ShowDoc" default="false"/>
    <param name="SavDoc" default="" type="string"/>
    <param name="SavFmt" default="*XLS"/>
    <param name="XlsOpt" default=""/>
    <param name="XlsMap" default=""/>
  </param1>
</command>
    
```

| Paramètres | Description |
|--------------------------------|---|
| String ToXls | Nom du document modèle. <u>Exemple:</u> C:\\temp\\template.xlsx |
| String ToSheet | Nom de la feuille Excel. |
| String ToName | Nom de la plage de destination. |
| String XlsOpt | Peut prendre les valeurs suivantes : <ul style="list-style-type: none"> • *REPLACE : les nouvelles données remplacent les anciennes données. • *RESIZE : les nouvelles données remplacent |

| | |
|-----------------------------------|---|
| | <p>les anciennes données. La zone dans le classeur est redimensionnée.</p> <ul style="list-style-type: none"> • *ADD : les nouvelles données sont ajoutées aux anciennes données. • *INSERT : des lignes sont insérées pour recevoir les nouvelles données. Le paramètre ENTIRER détermine si des lignes entières sont insérées, ou non. • *MAP : les valeurs des colonnes de la source de données sont affectées aux cellules portant les mêmes noms dans le tableau Excel. Seul le premier entregistrement de la source de donnée est utilisé. |
| String XlsMap | <p>Peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> • *MAPNAME : les valeurs des colonnes de la source de données sont affectées aux cellules portant les mêmes noms dans le tableau Excel. • *MAPVALUE : les valeurs des colonnes de la source de données sont affectées aux cellules dont la valeur initiale est identique |
| String ColPref | <p>Peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> • *NONE : aucun préfixe pour les colonnes • Valeur du préfixe pour les colonnes |
| String DataSource | Le path complet du fichier CSV. |
| String RecCnt | Si on utilise directement un fichier CSV comme source de données, il faut obligatoirement spécifier le nombre de lignes du fichier CSV. Dans tous les autres cas, cet attribut n'est pas utilisé. |
| String AutoFmt | <p>Peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> • *NONE : aucun format automatique n'est appliqué. • nom du format : le format spécifié est appliqué. |
| String AutoFit | <p>Peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> • True : ajuster automatiquement la largeur des colonnes. • False : ne pas modifier la largeur des colonnes. |
| String FmtCells | <p>Peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> • True : le format des cellules est fixé en fonction du type Base de Données. • False : le format des cellules est fixé par Excel, en fonction des valeurs. |
| String AddColH | <p>Peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> • True : les noms des colonnes sont insérés en première ligne. • False : les noms des colonnes ne sont pas insérés en première ligne. |
| String ShowDoc | <p>Peut prendre les valeurs suivantes :</p> <ul style="list-style-type: none"> • True : le document généré est visible • False : le document généré n'est pas visible |
| String SavDoc | Chemin complet du fichier résultat. <u>Exemple:</u> C:\\temp\\result.xlsx |
| String SavFmt | Peut prendre les valeurs suivantes : |

| | |
|--|---|
| | <ul style="list-style-type: none">• *NORMAL : format du fichier modèle• *XLS : format Excel 2003 |
|--|---|

Authentification

Les clients qui souhaitent utiliser les méthodes proposées par le web service Launcher REST Gateway doivent d'abord **s'authentifier en faisant appel à la méthode open**. Cette authentification s'opère par le biais d'Active Directory.

Une fois que le client est authentifié auprès du web service suite à un appel réussi à la méthode **open**, il recevra un **ID de client**, lui permettant ainsi d'appeler les autres méthodes du service qui nécessitent cet ID.

Le serveur Launcher auquel est connecté le client peut aussi nécessiter une authentification LDAP ([si configuration effectuée](#)), le client recevra le message « LDAP authentication required » lorsqu'il essaie d'exécuter les autres méthodes du web service.

Pour s'authentifier sur le serveur Launcher, le client doit alors appeler la méthode **ldapauth** après avoir appelé la méthode **open**.

La configuration de la méthode d'authentification de Launcher REST Gateway et d'autres détails sont précisés dans la partie [Configuration](#).

Utilisation depuis un client PHP

Quelques exemples d'utilisation sont donnés dans le dossier « **Exemples\PHP** ».

Le code qui prépare les messages pour l'envoi au web service peut être encapsulé dans une classe PHP (**Exemples\PHP\Classe\LauncherClient.PHP**) pour simplifier les appels aux méthodes du web service (testClient.PHP, testClient_LNCDBGET_LNCPRTDOC.php et testClient_LNCDBGET_LNCTOxls.php).

Autres exemples

Windev21

Un exemple complet et un guide sont fournis dans le répertoire "C:\Program Files (x86)\LAUNCHER XPRESS\Exemples\Windev".

Visual Studio 2015 - C#

Un exemple complet et un guide sont fournis dans le répertoire "C:\Program Files (x86)\LAUNCHER XPRESS\Exemples\VisualStudio_CSharp".

Programmation avancée

Les valeurs spéciales

Lors de l'utilisation des commandes LAUNCHER Office ou des programmes API, les paramètres peuvent accepter des valeurs spéciales, interprétées par LAUNCHER.

Ces valeurs spéciales permettent de désigner des propriétés de l'environnement Windows ou de LAUNCHER.

- **%LNCDIR%**

Cette valeur spéciale désigne le répertoire depuis lequel le serveur LAUNCHER a été lancé sur le PC.

Par défaut, ce répertoire est « C:\Program Files\LAUNCHER<X> ».

Exemple :

CHGVAR VAR(&PARM1) VALUE('%LNCDIR%\Samples\Model.dot')

- **%CHR(n)**

Permet de représenter un caractère par sa valeur ASCII.

n est un nombre décimal de 0 à 255.

Exemple :

CHGVAR VAR(&PARM1) VALUE('Ligne 1%CHR(10)Ligne2')

%CHR(10) insère un caractère « Saut de ligne ».

- **%PARA% ou %LF%**

Permet d'insérer un saut de paragraphe dans un texte Word ou un message électronique par exemple.

Exemple :

CHGVAR VAR(&PARM1) VALUE('Paragraphe 1%PARA%Suite')

- **%EURO%**

Insère le symbole EURO (€) dans le texte.

Exemple :

CHGVAR VAR(&PARM1) VALUE(&VAL *CAT ' %EURO%')

- **%TAB%**

Insère le caractère **Tabulation** dans le texte.

- **%CONID%**

Insère le numéro d'identification de la connexion LAUNCHER en cours. Ce numéro est unique.

Il est compris entre 1 et le nombre maximum de connexions simultanées acceptées.

Cet identifiant est utile pour générer des noms de fichiers temporaires et uniques.

- **%NONE%**



Rien. Initialiser PARM2 à %NONE% si on veut vider le contenu d'un bookmark.

- **%TEMP%**

Si le mot entre les caractères % est une variable de l'environnement de Windows, remplacé par sa valeur.

Exemple:

CHGVAR VAR(&PARM1) VALUE('%TEMP% \fichier.txt')

%TEMP% désigne le répertoire temporaire de Windows, tel qu'il est défini dans l'environnement.

- **%SEP%**

Pour passer à la cellule suivante.

Si on était positionné sur la dernière colonne du tableau, on passe au début de la ligne suivante.

Exemple:

CHGVAR VAR(&PARM1) VALUE('%INS%Valeur colonne 1%SEP%Valeur Colonne 2 & 3%MRG%%SEP%12300%SEP%')

- **%INS%**

Pour insérer une nouvelle ligne et se positionner en début de la nouvelle ligne.

Exemple:

CHGVAR VAR(&PARM1) VALUE('%INS%Valeur colonne 1%SEP%Valeur Colonne 2 & 3%MRG%%SEP%12300%SEP%')

- **%MRG%**

Pour fusionner la cellule courante avec la suivante.

Exemple:

CHGVAR VAR(&PARM1) VALUE('%INS%Valeur colonne 1%SEP%Valeur Colonne 2 & 3%MRG%%SEP%12300%SEP%')

- **%CRLF%**

Permet d'insérer un retour à la ligne dans une même cellule Word.

Word

Launcher Office et Word

Launcher permet de piloter toutes les versions de Word depuis Office 2010 jusqu'à Office 2019.

Pour insérer des données d'une base de données externe dans un document Word, le plus simple est d'utiliser la commande **LNCPRDOC**.

Si cette commande ne vous permet pas de d'atteindre le niveau de détails que vous souhaitez, utilisez les commandes **LNCOPEN**, **LNCCMD** et **LNCCLOSE**.

Pour utiliser Word depuis un programme AS/400, il faut suivre la procédure suivante :

```
/* Déclarer les variables du programme, et les initialiser */
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50) VALUE('*ONLY')
DCL VAR(&SVRADD) TYPE(*CHAR) LEN(30) VALUE('*DEV')
DCL VAR(&CCSID) TYPE(*CHAR) LEN(10) VALUE('*JOB')
DCL VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL VAR(&RESULT) TYPE(*CHAR) LEN(512)

/* Ouvrir la communication entre le PGM AS/400 et le PC */
CALL PGM(LNCOOPEN) PARM(&HANDLE &SVRADDR &CCSID)

/* Ouvrir Word, et, optionnellement, un modèle de document */
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE('New')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &OPT &CMD &PARM1 &PARM2 &RESULT)

/* Poursuivre la composition du document */
/* . . . */

/* Sauver, Imprimer, Afficher le document (Optionnel) */
/* . . . */

/* Fermer Word et le document (Optionnel) */
CHGVAR VAR(&CMD) VALUE('WORDCLOSE')
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &OPT &CMD &PARM1 &PARM2 &RESULT)

/* Fermer la communication */
CALL PGM(LNCCLOSE) PARM(&HANDLE)
```

Pour la composition du document, on peut utiliser la technique des "**signets**", ou la technique du "**Publipostage**".

Composition d'un document par les signets

Pour la composition du document, vous pouvez utiliser la technique des **signets**.

La technique des signets permet de composer ou de modifier un document Word, en adressant des positions du document pour y insérer du texte, des objets ou des images.

Un signet est un emplacement nommé dans un document Word.

Il peut représenter une position ou une sélection de plusieurs caractères.

Quand du texte est inséré devant une position désignée par un signet, le signet est aussi repoussé vers la fin du document, au fur et à mesure de l'insertion.

Un signet est inséré dans un modèle de document par le menu **Insertion – Signets**.

Si le signet doit désigner une sélection de texte, sélectionnez le texte, puis allez à :

Menu "**Insertion - Signets**".

Vous devez donner un nom au signet que vous insérez.

Ce nom sera utilisé dans le programme AS/400.

Depuis le programme AS/400, vous pourrez insérer du texte à l'emplacement du signet.

Exemple : Le contenu de la variable **&NOMCLI** est inséré à l'emplacement du signet nommé **SNOMCLIENT**.

```
CHGVAR VAR(&CMD) VALUE('WBOOKMARK ')  
CHGVAR VAR(&PM1) VALUE('SNOMCLIENT')  
CHGVAR VAR(&PM2) VALUE(&NOMCLI)  
CALL PGM(LNCCMD) PARM(&HDL &CMD &OPT &PM1 &PM2 &RES)
```

Si le signet représente une sélection, toute la sélection sera remplacée par le nouveau texte.

Pour vous aider à voir l'emplacement de signets, rendez les visibles :

Menu "**Outils - Options**"

Onglet "**Affichage**".

Dans la rubrique "**Afficher**", cochez "**Signets**".

Word affichera les symboles [**et**] en début et fin des signets.

Le mot clé de LAUNCHER [WBOOKMARK](#) sert à placer le curseur de Word, à l'emplacement du signet.

Après l'insertion d'un texte par le mot clé [WBOOKMARK](#), le curseur est placé à la suite de ce nouveau texte inséré.

L'utilisation du mot clé [WTYPETEXT](#) permet d'insérer du texte à la position courante du curseur.

Au cours de l'insertion de textes par [WBOOKMARK](#) ou [WTYPETEXT](#), Word remet en page le document, gère les sauts de lignes, les cadrages, les alignements et les sauts de pages.

Un signet peut aussi être utilisé pour placer le curseur en vue d'une insertion d'un autre document.

Voyez les mots clés [WINSERTE](#), [WINSERTIMG](#).

Des valeurs spéciales de noms de signets permettent de naviguer dans les tableaux, et d'y insérer des cellules. Voir le détail de [WBOOKMARK](#).

Word met à la disposition des programmeurs une série de signets prédéfinis pour atteindre certaines parties identifiables d'un document, tel que le début, la fin.

Voir la liste des signets prédéfinis.

Les signets peuvent se trouver dans n'importe quelle partie du document, entêtes et pieds de pages compris.

Attention, si un modèle de document comporte des signets, s'il est utilisé par un publipostage avec [LNCPRDOC](#), le résultat après publipostage n'a plus les signets.

Si vous devez combiner « Signets » et « Publipostage », votre programme AS/400 doit :

- Charger le modèle par un CALL [LNCCMD](#), mot clé [WORDOPEN](#),
- Modifier le document par les signets,
- Lancer le publipostage à partir du document courant, modifié.

Signets prédéfinis

Des signets sont prédéfinis dans Microsoft Word.

Ils sont utilisables dans toutes les commandes de LAUNCHER où un nom de signet est utilisable.

| Signet | Description |
|-----------|--|
| \Sel | Sélection courante ou point d'insertion. |
| \PrevSel1 | Sélection la plus récente où l'édition se produit. |

| | |
|---------------|---|
| \PrevSel2 | Deuxième sélection la plus récente où l'édition se produit. |
| \StartOfSel | Début de la sélection courante. |
| \EndOfSel | Fin de la sélection courante. |
| \Line | Ligne courante ou première ligne de la sélection courante. Si le point d'insertion est à la fin de la ligne qui n'est pas la dernière ligne du paragraphe, le signet inclus entièrement la prochaine ligne |
| \Char | Caractère actuel, qui est le caractère suivant le point d'insertion s'il n'y a aucune sélection, ou premier caractère de la sélection. |
| \Para | Paragraphe courant, qui est le paragraphe contenant le point d'insertion ou, si plus d'un paragraphe est sélectionné, le premier paragraphe de la sélection. Notez que si le point d'insertion ou la sélection est dans le dernier paragraphe du document, le signet " \Para " n'inclut pas la marque du paragraphe. |
| \Section | Section courante, incluant la rupture à la fin de la section. La section courante contient le point d'insertion ou la sélection. Si la sélection contient plus d'une section, le signet " \Section " est la première section dans la sélection. |
| \Doc | Contenu entier du document actif, avec l'exception de la marque finale de paragraphe. |
| \Page | Page courante, incluant la rupture à la fin de la page. La page courante contient le point d'insertion. Si la sélection courante contient plus d'une page, le signet " \Page " est la première page de la sélection. Notez que le point d'insertion ou la sélection est dans la dernière page du document, le signet " \Page " n'inclut pas la marque finale du paragraphe. |
| \StartOfDoc | Début du document. |
| \EndOfDoc | Fin du document. |
| \Cell | Cellule courante dans une table, qui est la cellule contenant le point d'insertion. Si une ou plusieurs cellules d'une table sont incluses dans la sélection courante, le signet " \Cell " est la première cellule dans la sélection. |
| \Table | Table courante, qui est la table contenant le point d'insertion ou la sélection. Si la sélection inclut plus d'une table, le signet " \Table " est la première table entière de la sélection, même si la table entière n'est pas choisie. |
| \HeadingLevel | L'entête qui contient le point d'insertion ou la sélection, plus quelques entêtes et texte subalternes. Si la sélection courante à un corps texte, le signet " \HeadingLevel " inclut l'entête précédente, plus n'importe quels entêtes et subalterne des textes à cette entête. |

Guide de Procédures pour Microsoft Word

Le Publipostage Word avec LNCPRDOC

Principe du publipostage de Word

Pour insérer des données de la base de données AS/400 dans un document Word, le plus simple est d'utiliser la commande CL de haut niveau de LAUNCHER Office **LNCPRDOC**.

Si cette commande ne permet pas de d'atteindre le niveau de détails souhaité, vous pouvez utiliser les programmes API [LNCOPEN](#), [LNCCMD](#) et [LNCCLDSE](#).

Principe du publipostage de Word :

Word **permet de générer un courrier à partir** d'un **modèle de publipostage**, des **données d'une base de données** ou d'un **fichier texte**.

Un **modèle de publipostage** est un document préparé.

En résumé, il contient :

- Les textes non variables,
- Les propriétés de mise en page,
- Les propriétés de formats des caractères,
- Des liens avec les champs de la base de données,
- Des expressions conditionnelles d'impression sur des mots ou des paragraphes.

Pendant la composition du courrier à imprimer, Word parcourt le fichier de base de données.

Pour chaque enregistrement, il « imprime » le contenu du modèle de document, en insérant les valeurs des champs de l'enregistrement en cours, aux emplacements où le modèle y fait référence.

Il tient compte des valeurs de l'enregistrement en cours sur les conditions d'impression.

Exemple :

CIVIL est le nom d'un champ de la base de données.

SI <CIVIL> = 1 ALORS "Monsieur" SINON "Madame" ,

Cette expression simplifiée imprimera « Monsieur » ou « Madame » en fonction de la valeur du champ CIVIL.

Un modèle de publipostage peut être une **lettre type** ou un **catalogue**.

Pour une **lettre type**, le modèle sert à générer un courrier pour chaque enregistrement de la base de données.

Pour un **catalogue**, le modèle est dupliqué à l'intérieur d'un seul courrier, autant de fois qu'il y a d'enregistrements.

Généralement, un modèle catalogue est constitué d'une seule ligne qui servira à présenter le contenu du fichier sous forme d'un tableau.

Généralement, le résultat d'un publipostage catalogue sera inclus à l'intérieur d'une lettre.

Composition d'un modèle de publipostage pour LAUNCHER Office

Pour réaliser votre modèle de publipostage, suivez la procédure suivante :

1) Créez une copie temporaire du fichier de base de données AS/400 sur le PC.

Ce n'est qu'une copie temporaire qui est réalisée, afin de connaître les noms des champs.

Les données transférées ne seront pas utilisées pour composer le courrier final.

Word a besoin d'un fichier de données en format TXT, CSV, MDB, ODBC, ... pour réaliser le publipostage.

LAUNCHER Office utilise par défaut le format TXT.

Il faut donc créer une copie du fichier en format TXT sur le PC.

Pour cela, depuis la session terminal, lancez la commande :

LNCXFER <F4>

Cette commande vous permettra de transférer le fichier ou la requête AS/400 vers votre répertoire de travail sur la PC.

Choisissez un nom de fichier PC avec l'extension ".TXT".

2) Ouvrez Word, créez un nouveau document vierge.

Quand Word est démarré, il crée un document vierge (Page blanche).

Vous pouvez lire un document déjà existant qui sera votre point de départ.

3) Transformez le document en modèle de publipostage.

Cliquez sur le menu de Word **Outils** » - « **Publipostage**.

Transformez le document principal en modèle lettre type ou catalogue.

4) Attachez le document à la source de données.

Toujours depuis l'assistant de publipostage de Word, vous devez "**ouvrir la source de données**".

Choisissez le fichier TXT que vous avez précédemment transféré par [LNCXFER](#).

Votre document est maintenant un modèle de publipostage.

5) Composez le document.

Quand le document est transformé en modèle de publipostage, et qu'une source de données est attachée, Word affiche des boutons complémentaires dans le menu.

Ces boutons vous aident à composer le document.

Ils vous aident à insérer des références aux champs de la base de données.

Tapez le texte du document.

Utilisez le bouton du menu Word pour insérer les champs de la base de données.

Pour insérer des expressions conditionnelles, utilisez le boutons du menu Word, ou le menu "**Insertion** - **Champ...**".

6) Sauvez le modèle de document.

Par le menu "**Fichier** - **Enregistrez sous**", sauvez le document sur un répertoire de votre disque ou du réseau.

7) Composez le courrier final.

Sur votre session AS/400, utilisez la commande :

LNCPRDOC <F4>.

Dans le paramètre DOC, désignez le modèle de document Word que vous venez de créer.

Avec les paramètre **MRGTYP**, **FROMFILE**, **FROMSQL** ou **QRYDFN**, choisissez le **fichier AS/400** à utiliser ou la **requête**.

Paramétrez la commande pour afficher, imprimer ou sauver le résultat du publipostage, et validez.

LAUNCHER Office envoie vers le PC le nouveau contenu du fichier AS/400, et lance la fusion avec le modèle Word.

8) Intégrez la commande LNCPRDOC dans votre chaîne de programmes.

```
PGM
LNCPRDOC DOC(S_Members.DOC) +
FLR('%LNCDIR%\SAMPLES') ROOT(*NONE) +
MRGTYPE(*FILE) FROMFILE(S_Members) +
OUTSAVE(*YES) SHOWDOC(*YES) EXESRV(*DEV) +
ENDOPT(*CON) +
SAVDOC('R_Members.doc') +
SAVFLR('%LNCDIR%\SAMPLES')
ENDPGM
```

Intégration de listes dans un courrier (Maître / Détail)

Word ne permet pas de réaliser un publipostage à partir de plusieurs sources de données.

Il faut donc le réaliser en autant d'étapes qu'il y a de fichiers.

Exemple :

Envoyer une lettre à un abonné, en rappelant la liste des produits empruntés.

L'identité de l'abonné vient d'un fichier.

La liste des produits empruntés vient d'un autre fichier, ou le nombre de lignes est variable.

A l'attention de Dupond

Adresse :

3 bld des Capucines
75018 Paris

Téléphone 0146133

Dupond Thierry

Veillez trouver ci dessus la liste des éléments actuellement en vot

| Titre | Objet | Auteur | Ge |
|---|--------|--------------|----|
| Astérix et Obélix, Mission Cléopatre | DVD | Alain Chabat | Av |
| Prisonnier Du Temps | K7 VHS | Jack Sholder | Av |

Nous vous rappelons que le délai maximum d'emprunt est fixé à de

Nous vous demandons donc de restituer le plus rapidement possibl

Je vous prie d'agréer, Dupond, mes salutations les meilleures.

1) Préparer un modèle de publipostage "Catalogue" pour la liste des produits.

Ce modèle est composé d'une seule ligne d'un tableau, dans lequel on insère la référence aux champs « **Référence** », « **Libellé** », « **Date** ».

Au cours du publipostage final, Word dupliquera cette ligne autant de fois qu'il y aura de factures pour le client.

L'exécution du publipostage final, par [LNCPRDOC](#), sauvera la liste sous le nom "Éléments empruntés.doc".



Utilisez les touches <ALT><F9> pour visualiser les expressions.



2) Préparer un modèle de publipostage "Lettre type" pour le courrier.

Insérez dans cette lettre les coordonnées du client, le texte, etc ...

A l'emplacement où la liste doit apparaître dans la lettre, insérez un champ :

Menu « **Insertion** » « **Champ ...** »

Choisissez **Liaison et Renvois** dans la liste de gauche,

et **INCLUDETEXT** dans la liste de droite.

Entrez le chemin vers le futur document "**Eléments empruntés.doc**", à la suite de INCLUDETEXT.

A l'attention de { MERGEFIELD NOM }
 { MERGEFIELD PRENOM }

Adresse :
 { MERGEFIELD NUMVOIE } { MERGEFIELD ADRESSE }
 { MERGEFIELD CODEPST } { MERGEFIELD CODEVILLE }

Téléphone { MERGEFIELD TELEPHONE }

{ IF="Mme" "Madame" "Monsieur" } { MERGEFIELD NOM } { MERGEFIELD PRENOM }

Veuillez trouver ci dessus la liste des éléments actuellement en votre possession.

| Titre | Objet | Auteur | Genre |
|--|-------|--------|-------|
| { INCLUDETEXT "Eléments empruntés.doc" * MERGEFIELD } | | | |

Nous vous rappelons que le délai maximum d'emprunt est fixé à deux semaines.

Nous vous demandons donc de restituer le plus rapidement possible ces ouvrages.

Je vous prie d'agréer, { IF="Mme" "Madame" "Monsieur" } { MERGEFIELD NOM } { MERGEFIELD PRENOM }
 salutations les meilleures.

3) Générer la liste des produits.

Lancez la commande [LNCPRTDOC](#) pour la liste.

```
LNCPRTDOC DOC('Modèle éléments +
empruntés.doc') FLR('%LNCDIR%\CSamples') +
ROOT(*NONE) MRGTYPE(*SQL) FROMSQL('SELECT +
```

```
* FROM S_EMPRUNTS WHERE NOADHR = 2') +
OUTSAVE (*YES) SHOWDOC (*NO) EXESRV (*DEV) +
ENDOPT (*NONE) SAVDOC ('Eléments +
empruntés.doc') +
SAVFLR ('%LNCDIR%€Samples') MRGSEL ('*CAT') +
WORDPATH ('%LNCDIR%€Samples') /* Premier +
publipostage de type Catalog */
```

Ce [LNCPRTDOC](#) transfère le fichier "liste de produits" vers le PC, Il génère un document intermédiaire, sauvé sous le nom "Eléments empruntés.doc".

4) Générer le courrier.

Lancez la commande [LNCPRTDOC](#) pour la lettre.

```
LNCPRTDOC DOC ('Modèle lettre de +
relance.doc') FLR ('%LNCDIR%€Samples') +
ROOT (*NONE) MRGTYPE (*SQL) FROMSQL ('SELECT +
* FROM S_MEMBERS WHERE NOADHR = 2') +
OUTPUT (*) OUTSAVE (*NO) SHOWDOC (*YES) +
EXESRV (*DEV) ENDOPT (*CON) +
WORDPATH ('%LNCDIR%€Samples')
```

Ce [LNCPRTDOC](#) transfère l'enregistrement du fichier "Abonné", génère le courrier final.

Le modèle Word utilisé inclus la liste "Eléments empruntés.doc".

Intégrez les deux appels à suivre dans votre CL.

Un même courrier peut être composé de plusieurs listes qui seront générées individuellement, avant d'être intégrées dans le courrier final.

Gestion des tableaux Word

L'utilisation des tableaux sous Word peut simplifier les cadrages des données.

Sous Word, au cours de la préparation du modèle, insérez un tableau par :

Menu « **Tableau** » « **Insérer** » « **Tableau** ».

Choisissez le nombre initial de lignes et colonnes.

Quand le curseur est placé à l'intérieur du tableau, vous avez accès à :

Menu « **Tableau** » « **Propriétés du tableau** ».

Vous pouvez choisir les encadrements, tailles, etc.

Une cellule d'un tableau est autonome.

Word gère les sauts de lignes et de paragraphes à l'intérieur de chaque cellule.

Avec LAUNCHER, si vous devez alimenter un tableau par la méthode des signets,

- Placez un signet à la première cellule que vous devez alimenter dans le tableau.
- Utilisez les mots clés [WBOOKMARK](#) pour vous placer sur cette première cellule.
- Utilisez le mot clé [WSETLINE](#) pour alimenter les valeurs de toutes les cellules d'une ligne.

Word insèrera des lignes au fur et à mesure de l'alimentation des valeurs de cellules.

L'exemple **S_11WORD** du fichier **QCLSRC** de la librairie **LAUNCHER** montre ce principe.

Comment créer un tableau sur plusieurs pages

Comment créer le document Modèle ?

LAUNCHER/400 Office

COMMANDE

Date :
N° de commande :

AURA Equipements
10, avenue du Québec
Bâtiment E1
BP 519
91946 LES ULIS CEDEX

| Référence | Quantité | Désignation | Prix Unitaire | Montant HT |
|-----------|----------|-------------|---------------|------------|
| | | | | |

Montant HT :
Total TTC :

See Conditions of Sales on Price list in Force

Explications

- Sélectionnez dans le menu Tableau, insérer Tableau ...
- Créer un tableau avec deux lignes
- Sur la première ligne du tableau insérer les titres des colonnes
- Sélectionner SEULEMENT la première ligne du tableau
- Tableau, Propriété du tableau ...
- Dans l'onglet ligne, cocher les deux options :
 - Autoriser le fractionnement des lignes sur plusieurs pages
 - Répéter en haut de chaque page en tant que ligne entête



Pour insérer une ligne en dessous, se positionner ABSOLUMENT sur la deuxième ligne du tableau

Insertion, zone de texte

Format de la zone de texte

Habillage : Autre

Bouton avancé : Position absolue pour Horizontal et vertical

Option, cocher :

Ancrer

Autoriser le chevauchement de texte

Résultat

Excel

Launcher et Excel

Launcher Office permet de piloter toutes les versions d'Excel depuis Office 2010 jusqu'à Office 2019.

Pour travailler avec Excel il faut avoir lancé Launcher Office sur le PC et sélectionné l'option "Démarrer Excel invisible" dans l'onglet configuration si vous voulez que Excel reste caché.

N'oubliez pas de fermer Excel à la fin de la session ([EXCELCLOSE](#)).

Mails

Envoi de mails

LAUNCHER Office permet d'envoyer des messages électroniques et d'en contrôler les principales caractéristiques :

- les destinataires (direct, en copie, en copie conforme)
- l'objet
- le texte du message
- la (les) pièce(s) jointe(s)
- le niveau de priorité
- l'activation de l'accusé de réception

Les messageries utilisées doivent être compatibles **MAPI** (ce qui est le cas de Outlook ou Exchange), sinon vous pouvez également utiliser [Lotus Notes](#) ou encore utiliser un [serveur SMTP](#).

Note : Pour envoyer vos messages, il n'est pas nécessaire que le système de messagerie soit en cours d'exécution sur le poste.

Les fonctions de gestion de mails

Certaines fonctions sont compatibles avec toutes les interfaces, d'autres sont dédiées à une interface particulière.

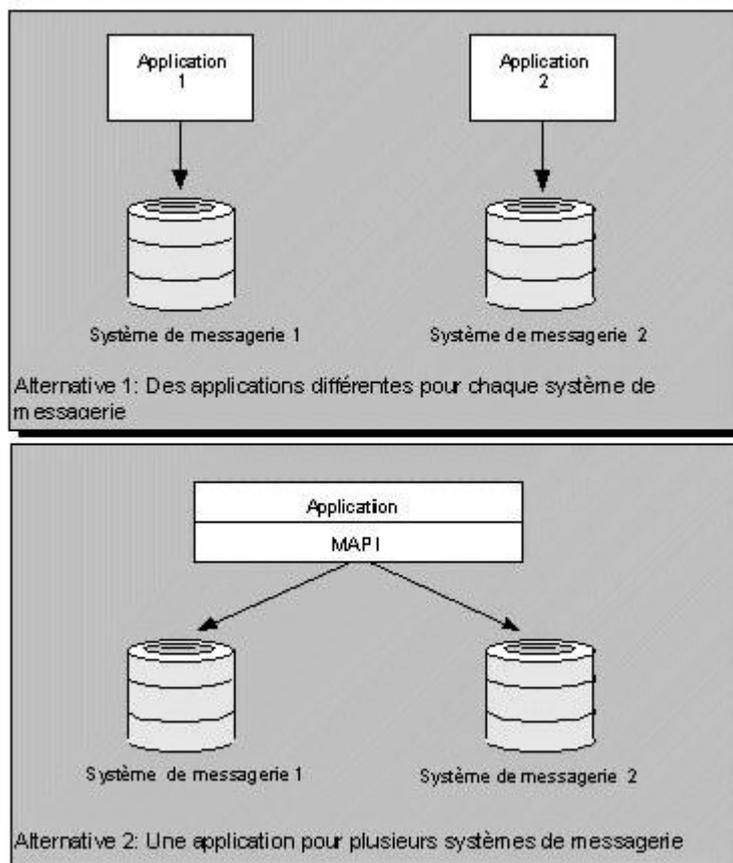
- : Fonctions compatibles avec les interfaces MAPI et Lotus Notes.
 - : Fonctions compatibles avec l'interface SMTP.
 - : Fonctions compatibles avec toutes les interfaces.
-
- [MAILPREP](#), pour préparer le message ;
 - [MAILATT](#), pour ajouter un fichier attaché au message ;
 - [MAILTEXT](#), pour définir un corps de texte ;
 - [MAILTO](#), ajoute des destinataires au message ;
 - [MAILPRTY](#), définition d'une priorité du message ;
 - [MAILSUBJ](#), ajoutant un sujet au message ;
 - [MAILREPORT](#), pour activer le suivi du message ;
 - [MAILSEND](#), envoie le message ;
 - [MAILEND](#), libère toutes les structures allouées par le message.
 - MAILSMTP, prépare les connexions avec le serveur SMTP;
 - MAILBODYF, permet d'envoyer un message à partir d'un page HTML
 - MAILCC, définit les destinataires secondaires du message.

La technologie MAPI

Pour pouvoir envoyer des e-mails, **LAUNCHER Office** utilise la technologie MAPI (Messaging Application Program Interface) développée par Microsoft.



Le principe de cette technologie est de proposer aux développeurs une "passerelle" pour leurs programmes afin qu'ils puissent fonctionner quelque soit le logiciel de messagerie utilisé (cf. page suivante pour les schémas).



De plus, en installant des plug-in spécifiques à chacun, vous pouvez envoyer des fax comme des e-mails.

Si vous voulez plus d'information, nous vous conseillons d'aller sur le site de connaissances de Microsoft (msdn.microsoft.com), et de consulter les articles concernant **Microsoft Fax**.

Pour Exchange, différents produits sur le marché existent.

Pour pouvoir utiliser les services d'Outlook ou Exchange avec **LAUNCHER Office**, il vous faut configurer ce dernier.

Ainsi, dans l'onglet configuration, vous trouverez une combo qui vous permet de choisir le type de messagerie.

Choisissez **MAPI**.

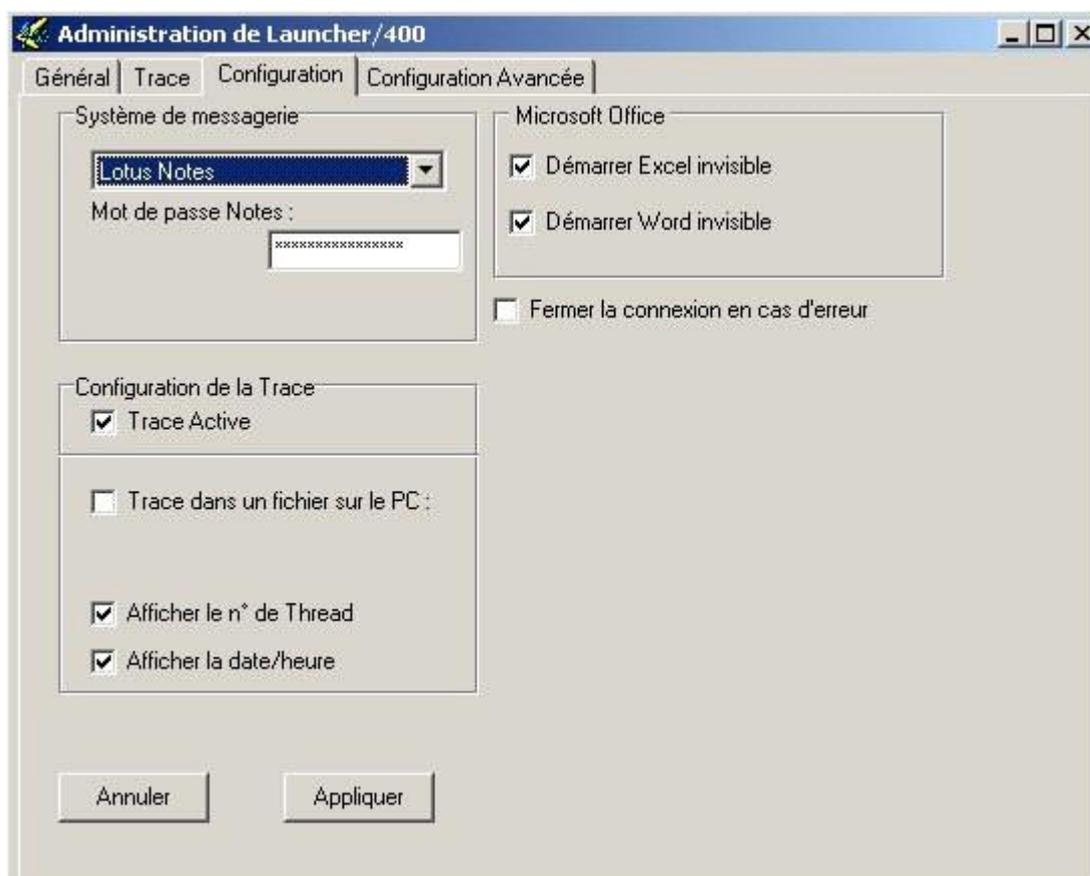
Vous pouvez également choisir le profil que vous allez utiliser lors de vos envois pendant toute la session de travail avec **LAUNCHER Office**.

Utilisation avec Lotus Notes

Lotus Notes n'est pas compatible avec MAPI (jusqu'à la version 5).

C'est pourquoi nous avons développé une interface spécifique à Notes, pour les versions 4.5 et 4.6. Cette interface vous permet d'utiliser les mêmes fonctions proposées par **LAUNCHER Office** que pour MAPI.

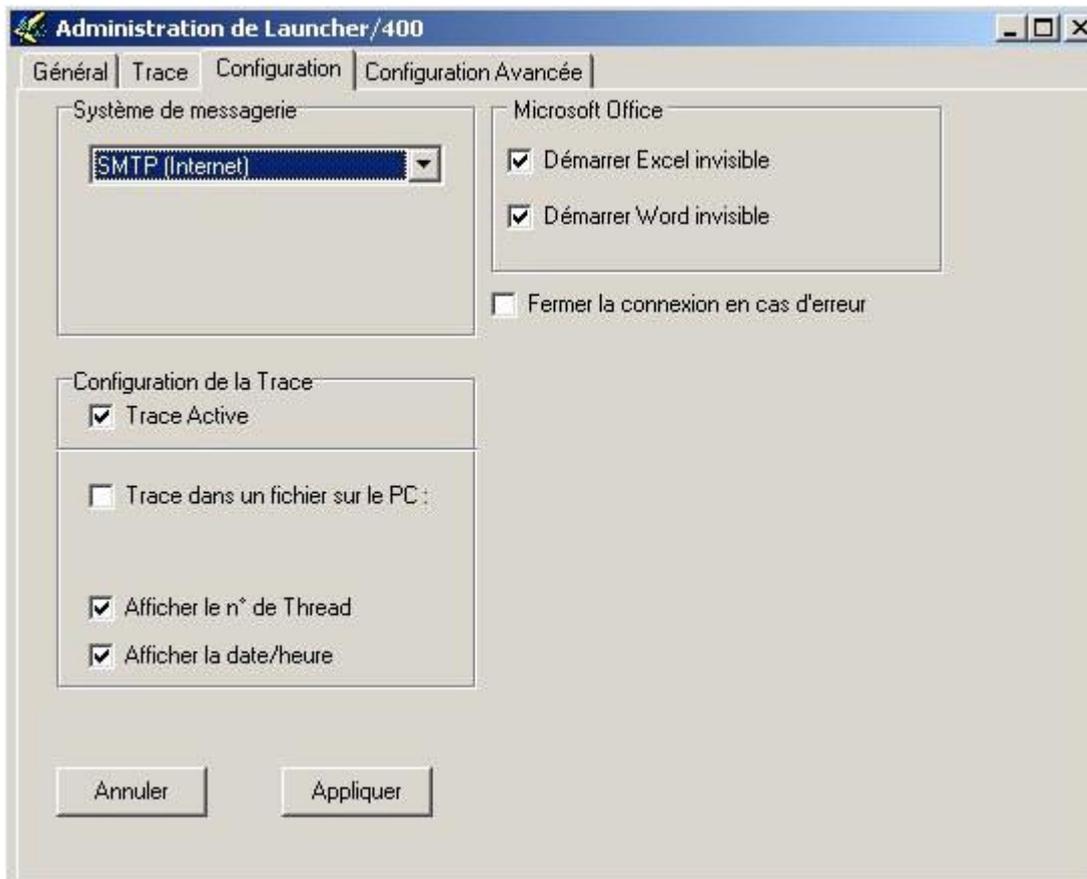
Pour utiliser Notes avec **LAUNCHER Office**, choisissez Lotus Notes comme type de messagerie, et indiquez votre mot de passe utilisateur Notes. Le profil est nécessairement celui du poste.



Utilisation avec un serveur SMTP

Si vous n'avez pas de messagerie compatible MAPI, mais que vous disposez d'un serveur SMTP, ou que vous souhaitez utiliser les facilités d'envois par SMTP, nous avons développé une interface spécifique SMTP.

Pour utiliser SMTP avec **LAUNCHER Office**, choisissez SMTP comme type de messagerie. Aucun profil n'est nécessaire, tout ce fait par programmation.



S/MIME

Depuis la version 2.7.1.1, les fonctionnalités S/MIME ont été introduites pour l'envoi de mails SMTP avec Launcher.

Nous nous sommes basés sur S/MIME Version 4.0, RFC 8551.

S/MIME (Secure/Multipurpose Internet Mail Extensions) est utilisé pour envoyer et sécuriser les données MIME.

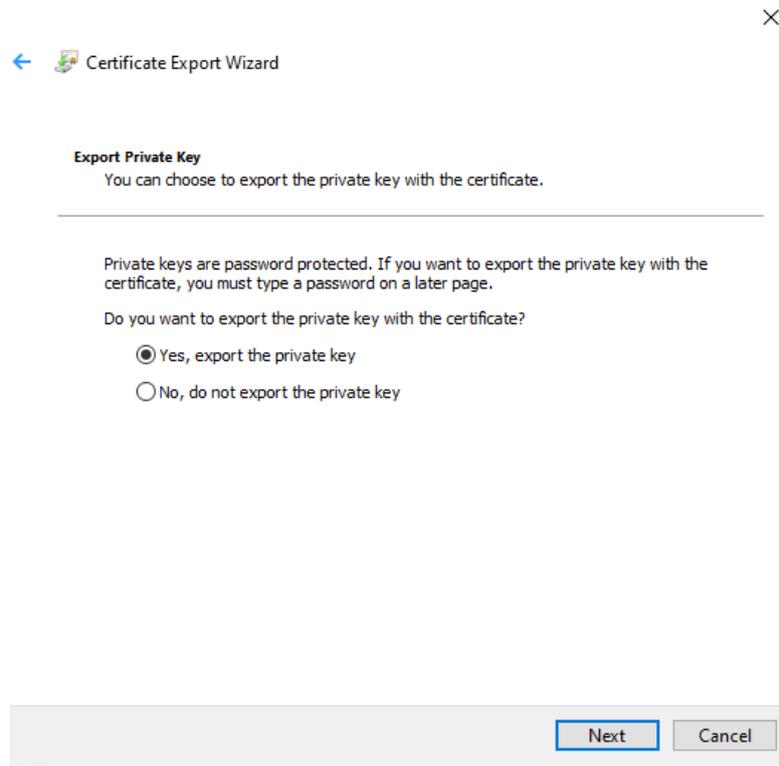
Basé sur la norme Internet MIME populaire, S/MIME fournit les services de sécurité cryptographiques suivants:

- authentification
- intégrité des messages et non-répudiation de l'origine (à l'aide de signatures numériques)
- confidentialité et sécurité des données (à l'aide du cryptage).

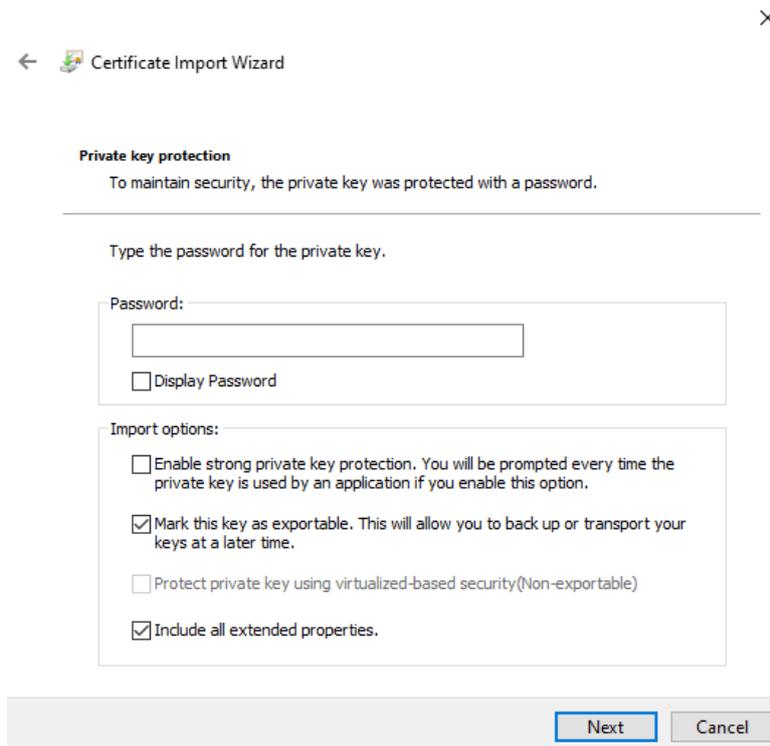
Certificat côté application cliente

Sur le PC/Serveur Windows envoyant des emails il est nécessaire d'avoir un certificat au format .p12 . Il s'agit d'un format binaire permettant de sauvegarder le certificat et

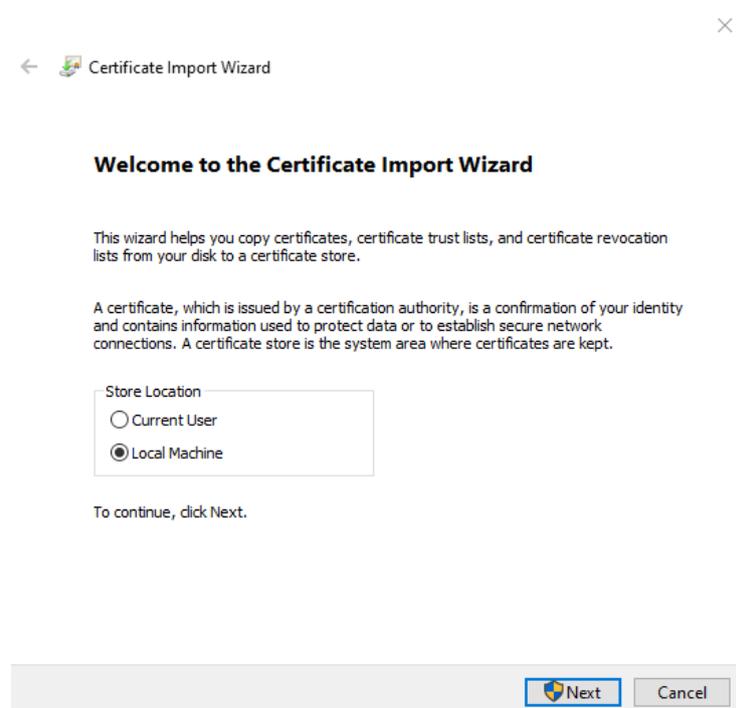
sa clé privée. Le fichier .p12 contient le certificat et la clé privée, et il est protégé par un mot de passe. Il peut être obtenu avec un export du certificat :



Si le fichier .p12 doit être installé sur un PC/Serveur Windows, et que vous voulez envoyer des emails signés avec Launcher, alors il faudra que lors de l'import du fichier .p12, "Mark this key as exportable" soit sélectionné :



De plus il faudra enregistrer le certificat dans "Local Computer", dans "Trusted Root Certification" :



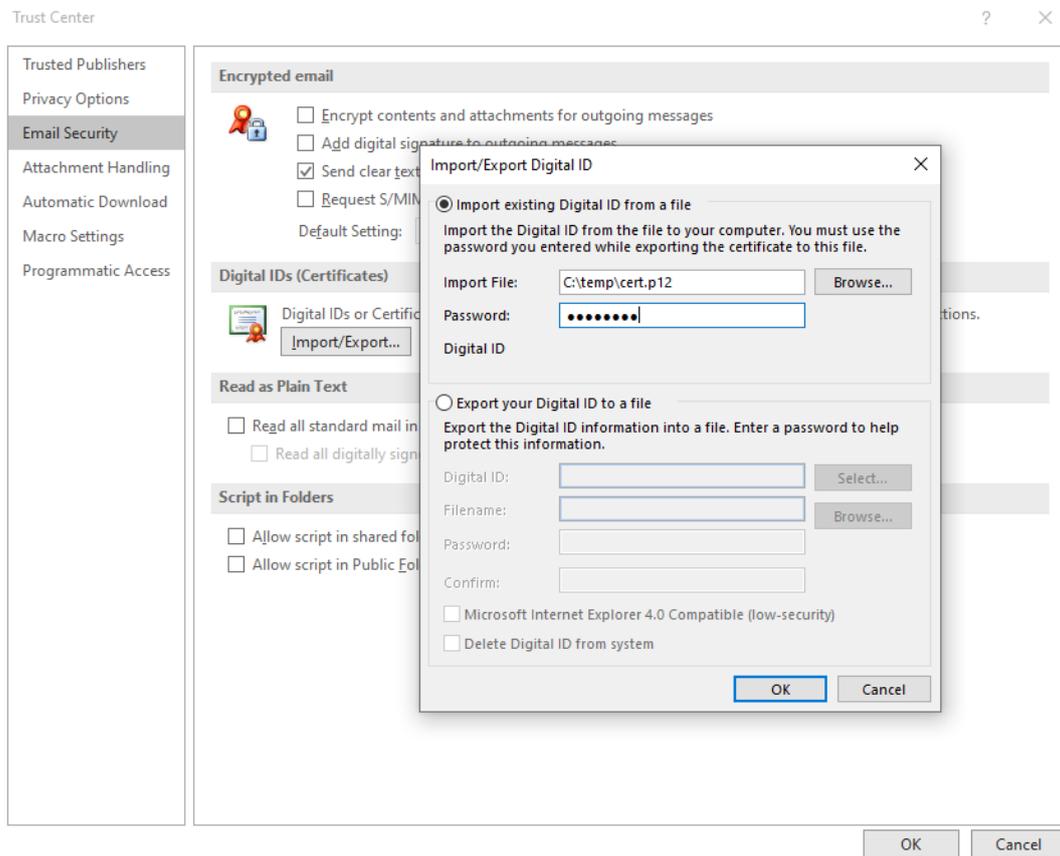
Les certificats installés dans "Local machine", peuvent être gérés avec : certlm.msc.

Certificat côté destinataire

Sur le PC/Serveur Windows du destinataire des emails, il faudra que le certificat soit installé.

Pour installer le certificat sur Outlook:

1. Ouvrez Outlook. Dans l'onglet Fichier, choisissez Options, puis Centre de confiance, puis Paramètres du centre de confiance.
2. Cliquez sur Email Security, puis sur Import / Export.
3. Cliquez sur Parcourir Recherchez votre fichier de certificat (.p12) et cliquez sur Ouvrir.
4. Saisissez le mot de passe utilisé pour sécuriser la clé privée, puis cliquez sur OK.



Les programmes API avec un client AS400

Déclaration des variables

Pour utiliser les programmes API de LAUNCHER Office en environnement OPM, les paramètres des programmes doivent être déclarés de façon stricte, en respectant les tailles décrites ci-après.

En environnement ILE, l'appel de la procédure passe "**L'operational Descriptor**", donc les tailles des paramètres passés en argument seront connues par la procédure appelée, et ne devront pas respecter strictement les tailles définies ci-après, à l'exception du paramètre 'HANDLE'.

Déclarations des variables dans le programmes AS/400:

Les programmes de LAUNCHER Office demandent des paramètres en entrée et sortie :

| Nom | Type | Taille | Description |
|--------|------|--------|--|
| HANDLE | CHAR | 50 | Identifiant de conversation entre le job et le PC. |
| SVRADD | CHAR | 30 | Nom ou adresse IP du PC à contacter. |
| CCSID | CHAR | 10 | CCSID AS/400 du travail. |
| CMD | CHAR | 10 | Verbe de la commande envoyée au PC. |
| OPT | CHAR | 1 | Option de commande. |
| PARM1 | CHAR | 512 | Paramètres de la commande |
| PARM2 | CHAR | 1024 | Paramètres de la commande (Compléments) |
| RESULT | CHAR | 512 | Résultat retourné par le PC. |

Voir aussi : [Commande PROPERTY](#)

Exemples de déclarations :

En CL :

```
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50) VALUE('*ONLY')
DCL VAR(&SVRADD) TYPE(*CHAR) LEN(30) VALUE('*DEV')
DCL VAR(&CCSID) TYPE(*CHAR) LEN(10) VALUE('*JOB')
DCL VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL VAR(&RESULT) TYPE(*CHAR) LEN(512)
```

En RPG ILE :

```
DHANDLE S 50 inz('*ONLY')
DSVRADD S 30 inz('*DEV')
DCCSID S 10 inz('*JOB')
DCMD S 10 inz('*DEV')
DOPT S 1 inz(*blanks)
DPARM1 S 512 inz(*blanks)
DPARM2 S 1024 inz(*blanks)
DRESULT S 512 inz(*blanks)
```

Toutes les commandes peuvent retourner un message d'erreur en utilisant le processus standard de gestion des erreurs de l'AS/400. Tous ces messages sont identifiés par un MSGID : **LNCnnnn**.

Exemple :

L'exemple ci-dessous ouvre une unique connexion (&HANDLE vaut *ONLY) sur le PC où se trouve l'émulateur qui l'appelle (utilisation du device *DEV) et avec le CCSID du Job en cours.

```
DCL                VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL                VAR(&SVRADDR) TYPE(*CHAR) LEN(30)
DCL                VAR(&CCSID) TYPE(*CHAR) LEN(10)

CHGVAR            VAR(&HANDLE) VALUE('*ONLY')
CHGVAR            VAR(&SRVADDR) VALUE('*DEV')
CHGVAR            VAR(&CCSID) VALUE('*JOB')

CALL              PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
MONMSG           MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Programmation ILE

Les procédures ILE de LAUNCHER Office sont définies dans le programme de service "LAUNCHER" de la librairie LAUNCHER.

Après la génération de votre module par le compilateur, créez votre programme exécutable, en le liant au programme de service:

```
CRTPGM PGM(MYPROG) MODULE(MYMODULE) ... BNDSRVPGM(LAUNCHER) ...
```

Lors de l'appel des procédures de LAUNCHER Office, choisissez les méthodes d'appel de votre langage de programmation, qui passent "**L'operational Descriptor**", à la procédure appelée.

En **CL**, "l'operational descriptor" est toujours passé:

```
CALLPRC PRC('LNCCMD') PARM(&HANDLE 'WORDOPEN' +  
*OMIT +  
'New;Visible=True' *OMIT &RESULT)
```

Avec **CALLB**, En RPG ILE, utilisez l'option (D):

```
CALLB(D) 'LNCCMD'  
PARM Handle  
PARM Cmd  
PARM Opt  
PARM Parm1  
PARM Parm2  
PARM Result
```

Avec **CALLP**, En RPG ILE:

```
D lncmd PR ExtProc('LNCCMD') OPDESC  
D handle 50 const options(*varsize)  
D cmd 10 const options(*varsize)  
D opt 1 const options(*varsize:*omit)  
D parm1 16000 const options(*varsize:*omit)  
D parm2 16000 const options(*varsize:*omit)  
D result 16000 options(*varsize:*omit)
```

```
c callp lncmd(handle:'SHELL':'0':  
c parm1:parm2:result)
```

Consultez les exemples dans les fichiers sources **QCLSRC** et **QRPGSRC** de la librairie LAUNCHER.

Utilisation dans un travail BATCH pour un client AS/400

LAUNCHER Office est utilisable dans des travaux BATCH.

La valeur spéciale ***DEV** n'est pas utilisable dans un travail BATCH pour désigner le PC avec lequel on veut communiquer.

Il faut ouvrir une conversation en désignant le PC par son nom ou son adresse IP.

Il est possible d'obtenir l'adresse IP du PC associé au terminal du travail interactif en cours, par le programme API [LNCOPEN](#) :

Voir l'exemple : [Récupérer le nom du profil utilisateur](#)

Utilisation dans un travail interactif

Dans un travail interactif, le programme AS/400 peut désigner le PC avec lequel il veut communiquer, en indiquant la valeur *DEV dans les paramètres « adresse du PC Serveur ».

LAUNCHER recherche alors l'adresse IP de l'écran associé au travail en cours.

Dans un travail interactif, les documents Word ou Excel peuvent être rendus visibles de l'utilisateur.

L'application s'ouvre alors, par dessus l'émulation de terminal en cours d'utilisation.

Des commandes LAUNCHER permettent de créer une interactivité entre le programme AS/400 et l'utilisateur.

On peut afficher des menus personnalisés, des boîtes de messages, des écrans d'exploration des répertoires de Windows, ou lancer des programmes sur le PC.

Le programme AS/400 peut se mettre en attente d'une action de l'utilisateur, avant de poursuivre son traitement.

LNCOPEN - Programme API

Ouvre la communication entre le travail AS/400 et un PC.

Avant l'appel à la fonction LNCOPEN, le démon **LAUNCHER Office** doit être lancé sur le PC ciblé.

Syntaxe

```

DCL      VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL      VAR(&SVRADDR) TYPE(*CHAR) LEN(30)
DCL      VAR(&CCSID) TYPE(*CHAR) LEN(10)

CHGVAR  VAR(&HANDLE) VALUE('*ONLY')
CHGVAR  VAR(&SRVADDR) VALUE('*DEV')
CHGVAR  VAR(&CCSID) VALUE('*JOB')

CALL    PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
MONMSG  MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|---------------|---|
| HANDLE (E/S) | <p>Descripteur retourné par la fonction LNCOPEN. Cette variable devra être utilisée dans toutes les fonctions de LAUNCHER Office.</p> <p>Initialiser la variable &HANDLE à *ONLY si vous n'ouvrez qu'une seule communication avec le PC.</p> <p>La valeur spéciale *GETDOT permet de récupérer l'adresse TCP/IP du PC dans le paramètre SVDADDR.</p> |
| SVRADDR (E/S) | <p>Nom réseau du PC ou adresse TCP/IP.</p> <p>Utiliser la valeur *DEV pour désigner le PC sur lequel tourne l'émulateur de terminal pour la session interactive en cour.</p> <p>Si le paramètre HANDLE vaut *GETDOT, SVRADDR contiendra l'adresse TCP/IP du PC en retour.</p> |
| CCSID (E) | <p>CCSID utilisé par le programme AS/400.</p> <p>Donnez la valeur *JOB pour utiliser le CCSID du job courant.</p> |

Remarque

Depuis la version V3R7 (CUMUL C8069370) de l'OS/400 il est possible d'utiliser les DEVICE AS/400 pour récupérer l'adresse IP de l'émulation.

En spécifiant ***DEV** pour l'adresse cible, le programme va converser avec le PC correspondant à l'émulation.

Pour utiliser LAUNCHER Office dans un travail « Batch », qui communique avec le PC de l'utilisateur, le programme interactif doit :

- Récupérer l'adresse du PC, avec HANDLE=*GETDOT, et SVRADDR=*DEV
- Soumettre le travail « Batch » en passant la valeur reçue dans le paramètre SVRADDR au programme soumis.

Exemple

Ouvrir une conversation avec le PC de l'utilisateur interactif.

```
DCL VAR(&SVRADDR) TYPE(*CHAR) LEN(30)
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL VAR(&CCSID) TYPE(*CHAR) LEN(10) VALUE(`*JOB`)
. . .
CHGVAR VAR(&SVRADD) VALUE('*DEV')
CHGVAR VAR(&HANDLE) VALUE(*ONLY')
CALL PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Ouvrir une conversation avec le PC à l'adresse "182.12.13.14"

```
. . .
CHGVAR VAR(&SVRADD) VALUE('182.12.13.14')
CHGVAR VAR(&HANDLE) VALUE(*ONLY')
CALL PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Récupérer l'adresse IP du PC de l'utilisateur interactif.

```
. . .
CHGVAR VAR(&SVRADD) VALUE(*DEV')
CHGVAR VAR(&HANDLE) VALUE(*GETDOT')
CALL PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
/* En retour, &SVRADDR contient l'adresse IP. */
```

LNCCMD - Programme API

Une fois la conversation ouverte avec [LNCOPEN](#) , **le programme LNCCMD permet l'envoi d'une commande au PC.**

C'est l'appel de programme principal que l'application répètera jusqu'à la fermeture de la connexion par [LNCCLOSE](#).

Syntaxe

```

DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)

CHGVAR      VAR(&CMD) VALUE('Commande')
CHGVAR      VAR(&OPT) VALUE('Option')
CHGVAR      VAR(&PARM1) VALUE('Premier_paramètre')
CHGVAR      VAR(&PARM2) VALUE('Second_paramètre')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1+
&PARM2 &RESULT)
MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | | |
|------------|---|--|
| HANDLE | E | Descripteur retourné par la fonction LNCOPEN ou *ONLY. |
| CMD | E | Mot clé de la commande à exécuter sur le PC. Voir Les mots clés de LNCCMD |
| OPT | E | Valeur de l'option selon la commande référencée par &CMD. OPT contient 'W' pour indiquer que PARM2 contient une valeur codée dans le jeu de caractère Unicode de Windows. |
| PARM1 | E | Ces paramètres sont affectés selon le mot clé spécifié dans le paramètre &CMD. Voir de détail de chaque mot clé : Les mots clés de LNCCMD |
| PARM2 | E | |
| RESULT | S | Résultat en retour, si la commande demandée retourne une valeur ou un message. |

Exemple

Lancer un programme sur le PC, et attendre son retour.



Pour cet exemple, consultez le détail de la commande [EXE](#) dans la liste alphabétique.

```
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50) VALUE('*ONLY')
DCL VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL VAR(&OPTION) TYPE(*CHAR) LEN(1)
DCL VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL VAR(&RESULT) TYPE(*CHAR) LEN(512)

CHGVAR VAR(&CMD) VALUE('EXE')
CHGVAR VAR(&OPT) VALUE('2')
CHGVAR VAR(&PARM1) VALUE('%LNCDIR%\SAMPLES\CALLEXE.EXE')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
```

Voir aussi

- [LNCOPEN](#)
- [LNCCLOSE](#)

LNCCLOSE - Programme API

Ferme la communication avec le PC.

Syntaxe

```
DCL          VAR (&HANDLE) TYPE (*CHAR) LEN (50)
```

```
CALL        PGM (LNCCLOSE) PARM (&HANDLE)
```

Paramètres

| Paramètres | |
|------------|---|
| HANDLE (E) | Descripteur retourné par la fonction LNCOPEN. |

Cette commande ferme la communication réseau entre l'AS/400 et le PC.

Exemple

```
CHGVAR      VAR (&HANDLE) VALUE ('*ONLY')
```

```
CALL        PGM (LNCCLOSE) PARM (&HANDLE)
```

Si l'on gère plusieurs connexions la variable &Handle contient leur identifiant, pour une connexion unique, utiliser *ONLY.

LNCDSCTS - Programme API

Le programme LNCDSCTS permet de créer une source de données (DSC : Data Source), à partir d'une structure de données en mémoire (DS).

Syntaxe

```

DCL          VAR(&DSC_NAME) TYPE(*CHAR) LEN(10)
DCL          VAR(&DS) TYPE(...)
DCL          VAR(&COUNT) TYPE(*DEC) LEN(5)
DCL          VAR(&OPTION) TYPE(*CHAR) LEN(10)
DCL          VAR(&DESC_FILE) TYPE(*CHAR) LEN(256)
DCL          VAR(&PROC_NAME) TYPE(*CHAR) LEN(256)

CALL        PGM(LNCDSCTS) PARM(&DSC_NAME &DS &COUNT +
                                &OPTION &DESC_FILE &PROCNAME)
    
```

Paramètres

| Paramètres | |
|------------|--|
| &DSC_NAME | Nom donné à la source de données. Ce nom sera utilisable par exemple lors de l'envoi des données vers un classeur Excel, avec la commande LNCTOxls : LNCTOxls ... CPYSRC(*DSC) FROMDSC(&DSC_NAME). |
| &DS | Variable de programme désignant le début du tableau de structures de données (DS) à référencer. En RPG, cette variable est le nom du tableau de DS. En CL, cette variable sera le premier membre composant la DS. |
| &COUNT | Nombre d'entrées dans le tableau de DS &DS. |
| &OPTION | Désigne de type de description fourni dans les paramètres &DESC_FILE et &PROC_NAME. Les valeurs possibles sont : *EXTNAME : La description des membres de la DS est donné par une DS externe. &DESC_FILE contient alors un nom de fichier physique, éventuellement qualifié sous la forme : LIB/FICHER. *PCML : La description est fournie par un fichier PCML. Le paramètre &DESC_FILE contient alors un chemin vers un fichier PCML dans l'IFS. &PROC_NAME est le nom d'une procédure présente dans le fichier PCML. |
| &DESC_FILE | Ce paramètre contient : <ul style="list-style-type: none"> • Un nom de fichier physique, si &OPTION=*EXTNAME.. |

- Un chemin vers un fichier PCML dans l'IFS, si &OPTION=*PCML.

&PROC_NAME Si &OPTION=*PCML, ce paramètre contient le nom de la procédure décrite dans le PCML. La description de cette procédure sera utilisée comme description des membres de la DS.

La conversation avec le serveur Launcher sur le PC doit être ouverte (par LNCOPEN) avant l'utilisation du programme ou de la procédure LNCDSCTS.

Les données du tableau de DS ne sont pas dupliquées. Seulement un pointeur vers la variable est pris.

La source de données est supprimée dès la fermeture de conversation par LNCLOSE, ou à la prochaine ouverture d'une nouvelle conversation.

Exemples

Dans l'exemple suivant, chaque occurrence du tableau de DS « CustList » est décrite par le fichier physique « S_CUST_DS ».

Le programme crée la source de donnée « MyDSC », et transfère les données contenues dans le tableau de DS « CustList » vers Excel par la commande LNCTOxls.

```

Dcustomer      E DS                      EXTNAME (S_CUST_DS)
...
dCustList      LIKEDS(customer) DIM(100)

C              CallB(D) 'LNCDSCTS'
C              Parm      'MyDSC'      DSCNAME
C              Parm      CustList
C              Parm      Count
C              Parm      '*EXTNAME'   Option
C              Parm      'S_CUST_DS'  DescFile
C              Parm      *BLANK       ProcName

C              eval      Qcmd = 'LNCTOxls TOxls(*NEW) -
c                      CPYSRC(*DSC) FROMDSC(MyDSC) -
c                      SHOWDOC(*YES) EXESRV(*CURRENT) -
c                      ADDCOLH(*COLHDG) '
C              CallP(e)  QCmdExc(Qcmd:%Len(%Trim(Qcmd)))
    
```

Dans l'exemple suivant, la DS a la même description que la structure « T_CRITSELPLANBBE » décrite dans le fichier PCML.

```

<pcml version="4.0">
  <struct name=="T_PLANBBE">
    ...
  </struct>
  <struct name=="T_CRITSELPLANBBE">
    <data name="NUMPLABBE" type="zoned" length="7" precision="0" usage="inherit" />
  />
  <data name="ORDPLABBE" type="char" length="3" usage="inherit" />
  <data name="NOMPLABBE" type="char" length="30" usage="inherit" />
    
```

```
<data name="IDPLANBBE" type="int" length="4" precision="31" usage="inherit" />
<data name="IDPLANBE" type="int" length="4" precision="31" usage="inherit" />
<data name="IDPLAIND" type="int" length="4" precision="31" usage="inherit" />
<data name="NOMFICBBE" type="char" length="50" usage="inherit" />
<data name="NOMINTBBE" type="char" length="50" usage="inherit" />
<data name="CHEMINBBE" type="char" length="3" usage="inherit" />
</struct>
</pcml>

c eval DescFile = '/Launcher/Descriptions.pcml'
c eval StrucName = 'T_CRITSELPLANBBE'

C CallB(D) 'LNCDSCLS'
C Parm 'MyDSC' DSCNAME
C Parm CustList
C Parm Count
C Parm '*PCML' Option
C Parm DescFile
C Parm ProcName
```

LNCWBM - Programme API

Le programme LNCWBM permet d'envoyer dans le document Word, les valeurs pour plusieurs signets, en un seul appel.

Syntaxe

```

DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&VAR1) TYPE( . . .
DCL          VAR(&VAR2) TYPE( . . .
DCL          VAR(&VARN) TYPE( . . .

CHGVAR      VAR(&PARM1) VALUE('noms des signets, et formats
           des variables')

CALL        PGM(LNCWBM) PARM(&HANDLE &PARM1 +
           &VAR1 &VAR2 . . . &VARN)

MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| &PARM1 | <p>Chaque signet dont la valeur est fixée par la commande est représenté entre parenthèses, selon la syntaxe :</p> <p>(NOM1:Fnn.dd)(NOM2:Fnn.dd)</p> <p>NOM1 et NOM2 sont les noms des signets. F est le type de la variable du programme AS/400 : A = Type caractère. P = Type décimal condensé (PACK). S = Type décimal étendu. nn est le nombre de chiffres ou de caractères de la variable. dd est le nombre de décimales.</p> <p>Si les valeurs à donner aux signets sont dans une variable DS, les signets sont alors représentés ainsi :</p> <p>DS:((NOM1:Fnn.dd)(NOM2:Fnn.dd)...) </p> |
| &VAR... | <p>Les variables contenant les valeurs à donner aux signets sont passées.</p> <p>Le nombre et l'ordre des variables correspond à celui décrit dans &PARM1.</p> |

Exemples

Dans l'exemple suivant, les variables du programme :

```
&ZPOLNV, &INTD, &NOMD, &IDADR1, &ZDVILL, &ACTIVITE
```

Contiennent chacune la valeur à affecter à un signet.

La variable &DS est une variable de 60 caractères au moins, et elle contient les valeurs à affecter aux deux signets : NOM (position 1 à 20) et ADQUERAB (position 21 à 60).

```
CHGVAR VAR(&PARM1) +  
VALUE(' (ZPOLNV:A10) (ZINTD:A20) (ZNOMD:A20) +  
(IDADR1:A20) (ZDVILL:A20) DS: ((NOM:A20) (ADQUERAB:A40)) +  
(ACTIVITE:P8.2) ' )  
CALL PGM(LNCWBM) PARM(&HANDLE &PARM1 &ZPOLNV &INTD &NOMD +  
&IDADR1 &ZDVILL &DS +  
&ACTIVITE)
```

LNCWFF - Programme API

Le programme LNCWFF permet d'envoyer dans le document Word, les valeurs pour plusieurs formulaires, en un seul appel.

Syntaxe

```

DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&VAR1) TYPE( . . .
DCL          VAR(&VAR2) TYPE( . . .
DCL          VAR(&VARN) TYPE( . . .

CHGVAR      VAR(&PARM1) VALUE('noms des formulaires, et
              formats des variables')

CALL        PGM(LNCWBM) PARM(&HANDLE &PARM1 +
              &VAR1 &VAR2 . . . &VARN)

MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| &PARM1 | <p>Chaque formulaire dont la valeur est fixée par la commande est représenté entre parenthèses, selon la syntaxe :</p> <p>(NOM1:Fnn.dd)(NOM2:Fnn.dd)</p> <p>NOM1 et NOM2 sont les noms des formulaires. F est le type de la variable du programme AS/400 : A = Type caractère. P = Type décimal condensé (PACK). S = Type décimal étendu. nn est le nombre de chiffres ou de caractères de la variable. dd est le nombre de décimales.</p> <p>Si les valeurs à donner aux formulaires sont dans une variable DS, les formulaires sont alors représentés ainsi :</p> <p>DS:((NOM1:Fnn.dd)(NOM2:Fnn.dd)...) </p> |
| &VAR... | <p>Les variables contenant les valeurs à donner aux formulaires sont passées.</p> <p>Le nombre et l'ordre des variables correspond à celui décrit dans &PARM1.</p> |

Exemples

Dans l'exemple suivant, les variables du programme :

&ZPOLNV, &INTD, &NOMD, &IDADR1, &ZDVILL, &ACTIVITE

Contiennent chacune la valeur à affecter à un formulaire.

La variable &DS est une variable de 60 caractères au moins, et elle contient les valeurs à affecter aux deux formulaires : NOM (position 1 à 20) et ADQUERAB (position 21 à 60).

```
CHGVAR VAR(&PARM1) +  
VALUE(' (ZPOLNV:A10) (ZINTD:A20) (ZNOMD:A20) +  
(IDADR1:A20) (ZDVILL:A20) DS: ((NOM:A20) (ADQUERAB:A40)) +  
(ACTIVITE:P8.2) ')  
CALL PGM(LNCWBM) PARM(&HANDLE &PARM1 &ZPOLNV &INTD &NOMD +  
&IDADR1 &ZDVILL &DS +  
&ACTIVITE)
```

LNCWVA - Programme API

Le programme LNCWVA permet d'envoyer dans le document Word, les valeurs pour plusieurs variables, en un seul appel.

Syntaxe

```

DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&VAR1) TYPE( . . .
DCL          VAR(&VAR2) TYPE( . . .
DCL          VAR(&VARN) TYPE( . . .

CHGVAR      VAR(&PARM1) VALUE('noms des variables, et formats
              des variables')

CALL        PGM(LNCWBM) PARM(&HANDLE &PARM1 +
              &VAR1 &VAR2 . . . &VARN)

MONMSG     MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    
```

Paramètres

| Paramètres | |
|------------|---|
| &PARM1 | <p>Chaque variable dont la valeur est fixée par la commande est représenté entre parenthèses, selon la syntaxe :</p> <p>(NOM1:Fnn.dd)(NOM2:Fnn.dd)</p> <p>NOM1 et NOM2 sont les noms des variables. F est le type de la variable du programme AS/400 : A = Type caractère. P = Type décimal condensé (PACK). S = Type décimal étendu. nn est le nombre de chiffres ou de caractères de la variable. dd est le nombre de décimales.</p> <p>Si les valeurs à donner aux variables sont dans une variable DS, les signets sont alors représentés ainsi :</p> <p>DS:((NOM1:Fnn.dd)(NOM2:Fnn.dd)...) </p> |
| &VAR... | <p>Les variables contenant les valeurs à donner aux variables sont passées.</p> <p>Le nombre et l'ordre des variables correspond à celui décrit dans &PARM1.</p> |

Exemples

Dans l'exemple suivant, les variables du programme :

```
&ZPOLNV, &INTD, &NOMD, &IDADR1, &ZDVILL, &ACTIVITE
```

Contiennent chacune la valeur à affecter à une variable.

La variable &DS est une variable de 60 caractères au moins, et elle contient les valeurs à affecter aux deux variables : NOM (position 1 à 20) et ADQUERAB (position 21 à 60).

```
CHGVAR VAR(&PARM1) +  
VALUE(' (ZPOLNV:A10) (ZINTD:A20) (ZNOMD:A20) +  
(IDADR1:A20) (ZDVILL:A20) DS: ((NOM:A20) (ADQUERAB:A40)) +  
(ACTIVITE:P8.2) ' )  
CALL PGM(LNCWBM) PARM(&HANDLE &PARM1 &ZPOLNV &INTD &NOMD +  
&IDADR1 &ZDVILL &DS +  
&ACTIVITE)
```

LNCCVTWCS - Programme API

Le programme LNCCVTWCS permet de convertir la contenu d'une variable chaîne de caractères depuis, ou vers, le jeu de caractères Unicode de Windows.

Syntaxe

```

DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&DIRECTION) TYPE(*CHAR) LEN(10)
DCL          VAR(&WCS_VAR) TYPE(*CHAR) LEN(...)
DCL          VAR(&WCS_SIZE) TYPE(*DEC) LEN(5)
DCL          VAR(&CHAR_VAR) TYPE(*CHAR) LEN(...)
DCL          VAR(&CHAR_SIZE) TYPE(*DEC) LEN(5)
DCL          VAR(&CCSID) TYPE(*DEC) LEN(5)
DCL          VAR(&RES_SIZE) TYPE(*DEC) LEN(5)
DCL          VAR(&OPTION) TYPE(*CHAR) LEN(...)

CALL         PGM(LNCCVTWCS) PARM(&HANDLE &DIRECTION +
                                &WCS_VAR &WCS_SIZE +
                                &CHAR_VAR &CHAR_SIZE +
                                &CCSID &RES_SIZE &OPTION)
    
```

Paramètres

| Paramètres | |
|------------|--|
| &DIRECTION | <p>Ce paramètre doit contenir :</p> <p>*TO pour convertir le contenu du paramètre &CHAR_VAR vers le jeu de caractère Unicode de Windows.</p> <p>*FROM pour convertir le contenu de &WCS_VAR depuis le jeu de caractères Unicode de Windows, vers un autre jeu.</p> |
| &WCS_VAR | <p>Entrée ou Sortie : Ce paramètre contient la chaîne de caractère en format Windows Unicode.</p> <p>Quand &DIRECTION=*TO, &WCS_VAR contient le <u>résultat</u> de la conversion.</p> <p>Quand &DIRECTION=*FROM, &WCS_VAR contient la chaîne à convertir.</p> |
| &WCS_SIZE | <p>Taille <u>en octets</u> du paramètre &WCS_VAR.</p> <p>Quand &DIRECTION=*TO, &WCS_SIZE représente la taille de la variable &WCS_VAR.</p> <p>Quand &DIRECTION=*FROM, &WCS_SIZE représente la taille à convertir dans &WCS_VAR. Si &WCS_SIZE est égal à -1, alors la chaîne contenue dans &WCS_VAR doit se terminer par un zéro.</p> |
| &CHAR_VAR | <p>Entrée ou Sortie : Ce paramètre contient la chaîne de caractères codée dans le jeu de caractère spécifié par le paramètre &CCSID.</p> |

| | |
|------------|---|
| | <p>Quand &DIRECTION='*TO', &CHAR_VAR contient la chaîne à convertir.</p> <p>Quand &DIRECTION='*FROM', &CHAR_VAR contient le résultat de la conversion.</p> |
| &CHAR_SIZE | <p>Taille <u>en octets</u> du paramètre &CHAR_VAR.</p> <p>Quand &DIRECTION='*TO', &CHAR_SIZE représente la taille à convertir dans &WCS_VAR.</p> <p>Quand &DIRECTION='*FROM', &CHAR_SIZE représente la taille de la variable &CHAR_VAR.</p> |
| &CCSID | <p>Jeu de caractère dans lequel est codé le contenu du paramètre &CHAR_VAR.</p> <p>Le contenu Unicode de &WCS_VAR sera converti depuis ou vers le jeu de caractère spécifié dans &CCSID.</p> <p>Si &CCSID vaut 0, c'est le CCSID du travail en cours qui est utilisé.</p> |
| &RES_SIZE | <p>Sortie. Taille en octets du résultat de la conversion généré dans &WCS_VAR ou &CHAR_VAR.</p> |
| &OPTION | <p>Les valeurs possibles sont :</p> <p>NZSTR : (Non zero string)</p> <p>Quand &DIRECTION='*TO', le résultat Unicode n'est pas terminé par un zéro de fin.</p> <p>Par défaut, un caractère 0 est ajouté à la fin de la chaîne Unicode, s'il n'y en avait pas dans la chaîne source.</p> <p>Quand &DIRECTION='*FROM', le caractère 0 éventuellement présent dans la chaîne Unicode, sera supprimé durant la conversion.</p> <p>Par défaut, si le caractère 0 est présent dans l'Unicode, il sera conservé et converti.</p> |

Le jeu de caractère Unicode Windows pris en compte par LNCCVTWCS **n'est pas** le jeu de caractère UCS2 de l'OS/400.

Utilisé avec les commandes [XLCELLS](#) et [WBOOKMARK](#) de LAUNCHER Office, ce programme LNCCVTWCS permet de lire ou d'écrire des textes dans Word ou Excel, dans des jeux de caractères externes (Chinois, Coréen, ...), et de gérer ces textes dans les polices DBCS correspondantes sur l'AS/400.

Exemples

```

/* Lire la valeur de la cellule B2 en Unicode Windows */
CHGVAR      VAR(&CMD) VALUE('XLCELLS')
CHGVAR      VAR(&PARM1) VALUE('Ref="$B$2";GetText=True;Unicode=True')
CHGVAR      VAR(&PARM2) VALUE(' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +

```

```
&PARM2 &RESULT)
/* &RESULT contient la valeur de B2 en Unicode Windows */
/* Terminé par un caractère 0 binaire */

/* Convertir &RESULT vers le CCSID chinois */
CHGVAR VAR(&WCS_LEN) VALUE(-1)
CHGVAR VAR(&CCSID) VALUE(835)
CHGVAR VAR(&CHINA_SIZE) VALUE(1000)
CALL PGM(LNCCVTWCS) PARM(&HANDLE '*FROM ' &RESULT &WCS_LEN +
&CHINA &CHINA_SIZE &CCSID &RES_SIZE)
```

LNCSETKW - Programme API

Le programme LNCSETKW permet d'insérer dans la variable &PARMX, une expression du type: 'MotCle=Valeur'. LNCSETKW facilite la construction des paramètres d'appel de LNCCMD. Si le mot clé existe déjà dans &PARMX, sa valeur sera remplacée.

Syntaxe

```

DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&PARMX) TYPE(*CHAR) LEN(512)
DCL          VAR(&KEYW) TYPE(*CHAR) LEN(...)
DCL          VAR(&TYPE) TYPE(*CHAR) LEN(10)
DCL          VAR(&VALUE) TYPE( . . .
DCL          VAR(&FORMAT) TYPE(*CHAR) LEN(10)

CALL         PGM(LNCSETKW) PARM(&HANDLE &PARMX +
                                &KEYW &TYPE &VALUE &FORMAT)
    
```

Paramètres

| Paramètres | |
|------------|---|
| &PARMX | &PARMX est la variable à modifier, qui sera utilisée lors du prochain appel à LNCCMD. En programmation OPM, &PARMX doit être une variable de type caractère , d'une taille de 512 . |
| &KEYW | &KEYW contient le mot clé à insérer dans &PARMX.. Le mot clé peut être encadré par des doubles quotes ("), ou, il doit se terminer par un caractère de ponctuation ou un blanc. Exemple: CHGVAR VAR(&KEYW) VALUE('Pattern;') Si &KEYW est blanc, alors la valeur de &VALUE sera ajoutée à &PARMX, précédée de ';' si &PARMX n'est pas vide. |
| &TYPE | &TYPE désigne le type de valeur attribuée au mot clé. Les valeurs possibles sont: * STR : Type chaîne de caractère; la valeur fournie par la paramètre &VALUE sera encadrée par des doubles quotes. * DEC : Type numérique; * BOO : Type booléen; &VALUE contient la valeur: '1' ou 'T' pour True, '0' ou 'F' pour False. * CAT : Type chaîne de caractère; La valeur fournie dans &VALUE sera concaténée à la valeur déjà présente pour le mot clé dans &PARMX. * CLR : Le mot clé et sa valeur sont supprimés de &PARMX, s'il existe. |
| &VALUE | Contient la valeur à attribuer au mot clé. Le type de &VALUE dépend de la valeur du paramètre &TYPE. |

| | |
|--------------------|--|
| &FORMAT | Format du paramètre &VALUE , sous la forme: Tn.d T est le type de donnée. Les valeurs possibles sont : A pour chaîne de caractères. P pour décimal condensé. S pour décimal étendu. n est le nombre de chiffres ou de caractères de la valeur, d est le nombre de décimales. |
|--------------------|--|

Exemple

Après l'exécution de la séquence d'instructions suivante :

```
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&KEYW) VALUE('Title ')
CHGVAR VAR(&TYPE) VALUE('*STR')
CHGVAR VAR(&VALUE) VALUE('Ouvrir le document')
CHGVAR VAR(&FORMAT) VALUE('A50')
CALL PGM(LNCSETKW) PARM(&HANDLE &PARM1 +
    &KEYW &TYPE &VALUE &FORMAT)

CALL PGM(LNCSETKW) PARM(&HANDLE &PARM1 +
    'FileMustExist' '*BOO ' '1')
```

La variable **&PARM1** aura la valeur:

```
'Title="Ouvrir le document";FileMustExist=True'
```

Voir aussi

[LNCGETKW - Programme API](#)

LNCGETKW - Programme API

**Le programme LNCGETKW permet de récupérer la valeur associée à un mot clé, en retour d'un appel à LNCCMD.
 Cette commande permet aussi de convertir un nombre stocké dans une variable chaîne de caractère vers un format décimal condensé ou étendu.**

Syntaxe

```
DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&PARMX) TYPE(*CHAR) LEN(512)
DCL          VAR(&KEYW) TYPE(*CHAR) LEN(...)
DCL          VAR(&VALUE) TYPE( . . .
DCL          VAR(&FORMAT) TYPE(*CHAR) LEN(10)

CALL         PGM(LNCGETKW) PARM(&HANDLE &PARM1 +
                                &KEYW &VALUE &FORMAT)
```

Paramètres

| Paramètres | |
|------------|---|
| &PARMX | <p>&PARMX est la variable retournée par un précédent appel à LNCCMD.</p> <p>En programmation OPM, &PARMX doit être une variable de type caractère, d'une taille de 512.</p> |
| &KEYW | <p>&KEYW contient le mot clé dont on veut récupérer la valeur dans &PARMX..</p> <p>Le mot clé peut être encadré par des doubles quotes ("), ou bien, il se termine par un des caractère de ponctuation ou un blanc.</p> <p><u>Exemple</u>: CHGVAR VAR(&KEYW) VALUE('Pattern;')</p> <p>Si le premier caractère de &KEYW est un astérisque (*), alors, les caractères suivants représente un format de stockage, sous la forme Tn.d, ceci pour convertir le texte stocké dans &PARMX, vers un format numérique dans &VALUE.</p> <p>T est un type de données de &VALUE: P pour décimal condensé, S pour décimal étendu. n et d désignent le nombre de chiffres et de décimales de la variable &VALUE.</p> <p><u>Exemple</u>: CHGVAR VAR(&KEYW) VALUE('*P8.2')</p> |
| &VALUE | <p><u>Résultat</u> : Variable destinée à recevoir la valeur associée au mot clé, ou le résultat de la conversion vers un format numérique.</p> |
| &FORMAT | <p>Format du paramètre &VALUE, sous la forme: Tn.d</p> <p>Ce format est donné ici, s'il n'a pas été donné par &KEYW.</p> <p>T est le type de donnée. Les valeurs possibles sont :</p> |

A pour chaîne de caractères.
P pour décimal condensé.
S pour décimal étendu.
n est le nombre de chiffres ou de caractères de la valeur,
d est le nombre de décimales.
Si **&FORMAT** est absent, la variable **&VALUE** est considérée comme étant de type caractère, et sa taille ne sera pas contrôlée.

Exemple

```
DCL VAR(&SBJ) TYPE(*CHAR) LEN(50)
DCL VAR(&SND) TYPE(*CHAR) LEN(50)
...
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 &PARM2 &RES)

/* Ici &RES contient: */
/* 'Subject="Mon message";SenderName="Service information"' */
CALL PGM(LNCGETKW) PARM(&HANDLE &RES 'Subject ' &SBJ 'A50')
CALL PGM(LNCGETKW) PARM(&HANDLE &RES 'SenderName ' &SND 'A50')

/* Après ces appels: */
/* &SBJ = 'Mon message' */
/* &SND = 'Service information' */
```

Voir aussi

[LNCSETKW - Programme API](#)

Programmes pour relier PC et AS400

Communication entre l'AS/400 et le PC

LAUNCHER Office met à disposition une bibliothèque de fonctions nommée LNCMsg qui permet à toute application PC de pouvoir envoyer des messages vers l'AS/400.

Toutes ces fonctions sont mises en œuvre dans l'exemple CALLEXE.

Pour appeler un programme sur le PC : [SHELL](#)

Les échanges de messages : [LNCMessage](#), [LNCRcvMessage](#), [LNCSetPostMessage](#)

LNCMessage

Permet d'envoyer un message vers l'AS/400, ou d'avertir l'AS/400 que le programme PC est terminé.

Prototype

```
long LNCMessage (long lRetCode, char *szRetString);
```

Paramètres

| Paramètres | |
|-----------------|--|
| IRetCode (E) | Code de retour interprété par LAUNCHER Office . |
| szRetString (E) | Message à envoyer à l'AS/400 |

Note

Avec un **IRetCode** égal à 0 ou à 1, utiliser le **szRetString** en lui affectant **"*END"**.

Voir aussi

- [LNCRcvMessage](#)
- [LNCSetPostMessage](#)

LNCRcvMessage

L'application se positionne en attente d'un message de l'AS/400.

Grâce à cette fonction, les paramètres fournis par l'AS/400 lors du deuxième appel, pourront être récupérés.

Prototype

```
long LNCRcvMessage (long lRetCode, char *szRetString);
```

Paramètres

| Paramètres | |
|-----------------|---------------------|
| lRetCode (S) | Code de retour. |
| szRetString (S) | Paramètres d'appel. |

Voir aussi

- [LNCMessage](#)
- [LNCSetPostMessage](#)

LNCSetPostMessage

Positionne un événement utilisable par LAUNCHER Office.

Prototype

```
long LNCSetPostMessage (long hHwnd,
                        long lMessage,
                        long wParam,
                        long lParam);
```

Paramètres

| Paramètres | |
|------------|-------------------------------|
| hHwnd | Fenêtre cible |
| lMessage | Message à envoyer |
| wParam | Premier paramètre du message |
| lParam | Deuxième paramètre du message |

Voir aussi

- [LNCMessage](#)
- [LNCRcvMessage](#)



Exemples de programmes CL

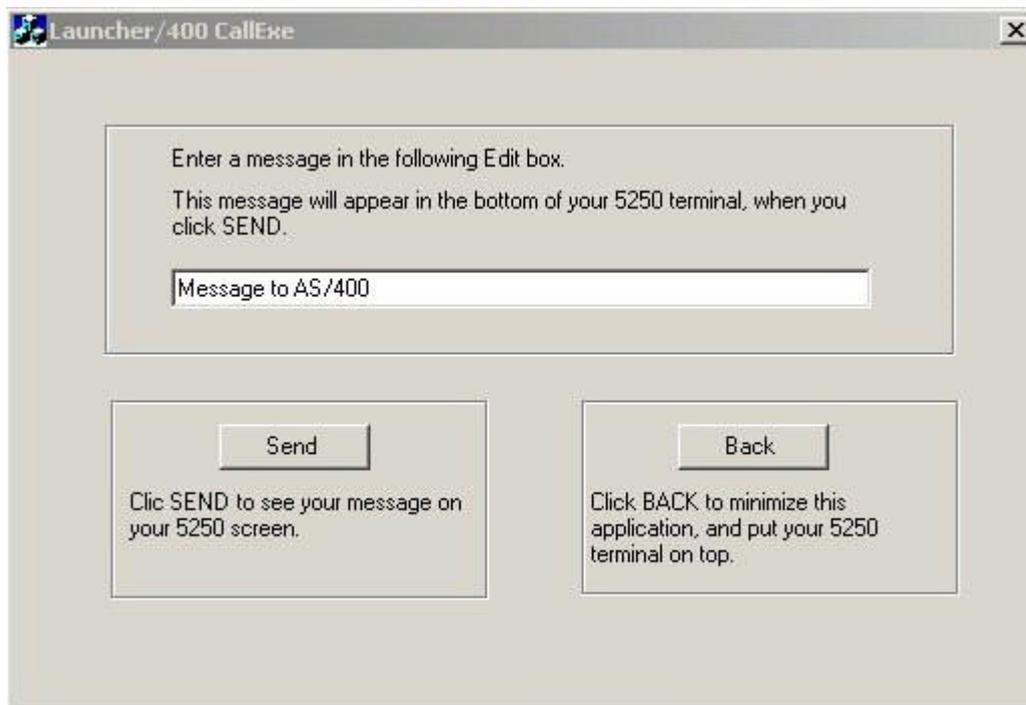
CallExe

Cet exemple permet d'envoyer des messages à l'AS/400.

Description

Lors de la réception du message *EXE*, le serveur exécute un programme constitué de la boîte de dialogue ci-dessous. **LAUNCHER Office** se met alors en attente d'un événement provenant de l'application.

Un clic sur le bouton prévu et le message est envoyé à l'AS/400 :



Lors du clic sur le bouton "Message to AS/400", le programme utilise la fonction LNCMessage se trouvant dans la bibliothèque LNCMSG.DLL et permet donc d'envoyer des messages à l'AS/400.

La fin du programme est provoquée par l'appui sur le bouton "Back To Terminal". L'AS/400 repasse alors en mode de commande, tandis que le PC passe en attente de connexion.

Sur l'AS/400 :

Paramètres ou commande
=> Call callexe '194.206.160.97'

Message To Send To The AS/400

Source du programme Visual Basic

Déclarations :

```
' Fonctions Launcher de LNCMSG.DLL
' -----
' Envoi un message à l'AS/400 et lui rend la main
Declare Function LNCMessage Lib "lncmsg.dll" (ByVal nRetCode As Long,
ByVal szRetString As String) As Long
' Reception de message depuis l'AS/400
Declare Function LNCRCvMessage Lib "lncmsg.dll" (piRetCode As Long,
ByVal szRetString As String) As Long
' Positionnement d'une référence à l'événement attendu de l'AS/400
Declare Function LNCSetPostMessage Lib "lncmsg.dll" (ByVal hwnd As
Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As Long)
As Long

' Fonctions Windows
' -----

' Get the ID of a control in the form.
Declare Function GetDlgCtrlID Lib "user32" (ByVal hwnd As Long) As
Long

Global Const WM_COMMAND As Long = &H111
Global Const BN_CLICKED As Long = 0

Global Const RET_REJECT = 0
Global Const RET_ACKNOWLEDGE = 1
Global Const RET_MESSAGE = 2
```

Sur le bouton Message To AS/400 :

```
Dim ret As Long
Dim RetCode As Long
Dim RetString As String * 100
Dim DlgId As Integer
' LAUNCHER/400 va simuler le clic sur le bouton ' 'Command3'
DlgId = GetDlgCtrlID(Command3.hwnd)
ret = LNCSetPostMessage(Form1.hwnd, WM_COMMAND, DlgId, Command3.hwnd)
' Envoie un message à l'AS/400.
ret = LNCMessage(2, Text1.Text)
```

Sur le bouton Back To Terminal :

```
Dim ret As Long
' Prévenir l'AS/400 de l'arret
ret = LNCMessage(2, "*END")
```

Sur le bouton Command3 (invisible) :

```
Dim ret As Long
```



```
Dim str As String * 200

' Ce bouton à été cliqué par LAUNCHER/400,
' lorsque l'AS/400 appelle le programme pour la
' deuxième fois et plus.
' Il est possible de récupérer un message
' de l'AS/400.
ret = LNCRCvMessage(ret, str)
Text1.Text = str
```

Lors de la première exécution, après le clic sur le bouton "**Message to AS/400**", LAUNCHER Office va simuler l'événement "**Clic sur le bouton Command3**", plaçant l'application en attente de message.

Une fois l'événement simulé, le programme envoie le texte du contrôle Text1 à l'AS/400.

Lorsque l'AS/400 appelle le programme pour la deuxième fois, le point d'entrée se fait dans le code du bouton caché (Command3), qui permet de récupérer les paramètres fournis par l'AS/400 à LAUNCHER pour l'exécution du programme.

Source du programme CLP

```
/* Variables utilisées pour ouvrir la communication*/
PGM          PARM(&SVRADDR)
DCL          VAR(&SVRADDR) TYPE(*CHAR) LEN(30)
DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&CCSID) TYPE(*CHAR) LEN(10)

/* Variables utilisées pour envoyer les commandes*/
DCL          VAR(&CMD)          TYPE(*CHAR) LEN(10)
DCL          VAR(&OPT)         TYPE(*CHAR) LEN(1)
DCL          VAR(&PARM1)       TYPE(*CHAR) LEN(512)
DCL          VAR(&PARM2)       TYPE(*CHAR) LEN(1024)
DCL          VAR(&RESULT)      TYPE(*CHAR) LEN(512)
DCL          VAR(&MSGID)       TYPE(*CHAR) LEN(7)
DCL          VAR(&MSG1)        TYPE(*CHAR) LEN(80)
DCL          VAR(&MSG2)        TYPE(*CHAR) LEN(80)
/* Variable spécifiant le protocole de communication */
DCL          VAR(&NETTYPE)     TYPE(*CHAR) LEN(6)
/* Ouverture d'une seule communication avec le PC*/
CHGVAR       VAR(&HANDLE) VALUE('*ONLY')
/* Traduit le CCSID selon le job CCSID */
CHGVAR       VAR(&CCSID) VALUE('*JOB')
/* Protocole de communication IP */
CHGVAR       VAR(&NETTYPE) VALUE('IP')

/* Connexion au PC */
CALL         PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* Sauve la fenêtre active sur le PC */
CHGVAR       VAR(&CMD) VALUE('STO')
CHGVAR       VAR(&PARM1) VALUE(' ')
CHGVAR       VAR(&PARM2) VALUE(' ')
```

```

CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG        MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/*Minimise fenêtre active (Emulation AS/400) */
CHGVAR        VAR(&CMD) VALUE('MIN')
CHGVAR        VAR(&PARM1) VALUE(' ')
CHGVAR        VAR(&PARM2) VALUE(' ')
CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG        MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* Début de la communication avec l'application CALLEXE.EXE, OPT=
/* donc le démon envoie un message a l AS 400 suite à un évènement */
/* de l'application */
CALLIT:
CHGVAR        VAR(&CMD) VALUE('EXE')
CHGVAR        VAR(&OPT) VALUE('2')
CHGVAR        VAR(&PARM1) VALUE('%LNCDIR%\SAMPLES\CALLEXE.EXE')
CHGVAR        VAR(&PARM2) VALUE(' ')
CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)

MONMSG        MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
SNDPGMMSG    MSGID(CPF9898) MSGF(QSYS/QCPFMSG) MSGDTA(&RESULT) +
                TOPGMQ(*EXT) MSGTYPE(*STATUS)
IF            COND(%SST(&RESULT 1 4) =
                CMDLBL(FINISH))
GOTO          CMDLBL(CALLIT)

FINISH:
/* Minimise la fenêtre active (Application CALLEXE) */
CHGVAR        VAR(&CMD) VALUE('MIN')
CHGVAR        VAR(&PARM1) VALUE(' ')
CHGVAR        VAR(&PARM2) VALUE(' ')
CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG        MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* Restaure la fenêtre stockée (Emulation AS/400)*/
CHGVAR        VAR(&CMD) VALUE('RCL')
CHGVAR        VAR(&OPT) VALUE('0')
CHGVAR        VAR(&PARM1) VALUE(' ')
CHGVAR        VAR(&PARM2) VALUE(' ')
CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)

/* Fin de la communication avec le PC */
CLOSE:
CHGVAR        VAR(&CMD) VALUE('END ')
CHGVAR        VAR(&PARM1) VALUE(' ')
CHGVAR        VAR(&PARM2) VALUE(' ')
CALL          PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                &PARM2 &RESULT)
MONMSG        MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

CALL          PGM(LNCCLOSE) PARM(&HANDLE)
MONMSG        MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
GOTO END
    
```

```
ERROR:
CALL          PGM(LNCCLOSE) PARM(&HANDLE)
RCVMSG       MSG(&MSG1) SECLVL(&MSG2) MSGID(&MSGID)
SNDPGMMMSG   MSGID(CPF9898) MSGF(QSYS/QCPFMSG) MSGDTA('Error +
              !!!!! ' *CAT &MSGID *CAT ' + ' *CAT &MSG1 *CAT ' +
              ' *CAT &MSG2) MSGTYPE(*ESCAPE)

END:
ENDPGM
```

Voir également l'exemple EVENTTEST dans la bibliothèque d'installation de l'AS/400 qui illustre l'échange de messages de synchronisation entre le PC et l'AS/400.

Exemple de fusion

Le programme MERGE est un petit exemple de publipostage.

Il utilise comme document principal le fichier sp_cust.doc du répertoire Samples de LAUNCHER Office et une source de donnée dynamiquement transférée par le programme MERGE de la bibliothèque LAUNCHER.

Dans un premier temps le programme utilise la commande [DBFXFER](#) pour transférer le fichier SP_CUST de la bibliothèque Launcher vers le PC.

Ensuite la commande [WMAILMERGE](#) lance la fusion vers un nouveau document.

Taper CALL LAUNCHER/MERGE pour l'exécuter.

Source de l'exemple

```
                PGM
/*  DECLARATIONS */
                DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
                DCL          VAR(&SVRADR) TYPE(*CHAR) LEN(30)
                DCL          VAR(&CCSID) TYPE(*CHAR) LEN(10)
                DCL          VAR(&CMD) TYPE(*CHAR) LEN(10)
                DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
                DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
                DCL          VAR(&PARM2) TYPE(*CHAR) LEN(1024)
                DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)
                DCL          VAR(&MSGID) TYPE(*CHAR) LEN(7)
                DCL          VAR(&MSG1) TYPE(*CHAR) LEN(80)
                DCL          VAR(&MSG2) TYPE(*CHAR) LEN(80)
/*  INITIALISATION */
                CHGVAR       VAR(&HANDLE) VALUE('*ONLY')
                CHGVAR       VAR(&SVRADR) VALUE('*DEV')
                CHGVAR       VAR(&CCSID) VALUE('*JOB')
/*  OUVERTURE DE LA SESSION - OPEN THE SESSION */
                CALL          PGM(LAUNCHER/LNCOOPEN) PARM(&HANDLE &SVRADR &CC
SID)
                MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/*  FILE TRANSFERT - TRANSFERT DU FICHER */
                CHGVAR       VAR(&CMD) VALUE('DBFXFER')
                CHGVAR       VAR(&PARM1) VALUE('%LNCDIR%\SAMPLES\SP_CUST.TX
T')
                CHGVAR       VAR(&PARM2) VALUE('LAUNCHER/SP_CUST')
                CALL          PGM(LAUNCHER/LNCCMD) PARM(&HANDLE &CMD &OPT &P
ARM1 +
                                &PARM2 &RESULT)
                MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/*  OUVERTURE WORD - OPEN WORD */
                CHGVAR       VAR(&CMD) VALUE('WORDOPEN')
                CHGVAR       VAR(&PARM1) VALUE(' ')
                CHGVAR       VAR(&PARM2) VALUE(' ')
                CALL          PGM(LAUNCHER/LNCCMD) PARM(&HANDLE &CMD &OPT &P
ARM1 +
                                &PARM2 &RESULT)
                MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/*  FUSION - MERGE */
```

```

                CHGVAR      VAR(&CMD) VALUE('WMAILMERGE')
                CHGVAR      VAR(&PARM1) VALUE('DOCUMENT=
                .DOC";DATASOURCE=
                CHGVAR      VAR(&PARM2) VALUE(' ')
                CALL        PGM(LAUNCHER/LNCCMD) PARM(&HANDLE &CMD &OPT &P
ARM1 +
                &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* WORDWAIT */
                CHGVAR      VAR(&CMD) VALUE('WORDWAIT')
                CHGVAR      VAR(&PARM1) VALUE(' ')
                CHGVAR      VAR(&PARM2) VALUE(' ')
                CALL        PGM(LAUNCHER/LNCCMD) PARM(&HANDLE &CMD &OPT &P
ARM1 +
                &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* FERMETURE SESSION LAUNCHER - CLOSE LAUNCHER SESSION */
                CHGVAR      VAR(&CMD) VALUE('END')
                CHGVAR      VAR(&PARM1) VALUE(' ')
                CHGVAR      VAR(&PARM2) VALUE(' ')
                CALL        PGM(LAUNCHER/LNCCMD) PARM(&HANDLE &CMD &OPT &P
ARM1 +
                &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
                CALL        PGM(LAUNCHER/LNCCLOSE) PARM(&HANDLE)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
                GOTO        END
ERROR:
                RCVMSG      MSG(&MSG1) SECLVL(&MSG2) MSGID(&MSGID)
                SNDFPMMSG   MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                MSGDTA('ERROR !!!!! ' *CAT &MSGID *CAT ' ' +
                ' *CAT &MSG1 *CAT ' ' *CAT &MSG2) +
                MSGTYPE(*ESCAPE)
END:
                ENDPGM
    
```

Exemple de mail

La commande MAILSEND est un programme en CL qui montre l'élaboration d'un mail avec un fichier attaché et un niveau de priorité élevé.

Vous pouvez éditer le source depuis le fichier QCLSRC de la bibliothèque LAUNCHER et remplacer les paramètres par défaut avec les vôtres.

Les valeurs par défaut sont les suivantes :

```
CHGVAR      VAR(&ADDR) VALUE('INFO@THECOMPAGNY.COM')
CHGVAR      VAR(&SUBJ) VALUE('THE CUSTOMER LIST')
CHGVAR      VAR(&MMSG) VALUE('Hello,%PARA%%PARA%Here +
                             is the list of the customers you +
                             required. The file comes from our AS/400. +
                             Isn''t this great ?%PARA%%PARA%See you +
                             soon !')
```

Source de l'exemple

```
/* VARIABLES USED TO OPEN COMMUNICATION AND SEND COMMANDS ...*/
PGM
DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50) VALUE('*ONLY
')
DCL          VAR(&SVRADDR) TYPE(*CHAR) LEN(30) VALUE('*DEV
')
DCL          VAR(&CCSID) TYPE(*CHAR) LEN(10) VALUE('*JOB'
)
DCL          VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)
DCL          VAR(&MSGID) TYPE(*CHAR) LEN(7)
DCL          VAR(&MSG1) TYPE(*CHAR) LEN(80)
DCL          VAR(&MSG2) TYPE(*CHAR) LEN(80)
/* VARIABLES FOR THE MAIL */
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(64)
DCL          VAR(&SUBJ) TYPE(*CHAR) LEN(64)
DCL          VAR(&MMSG) TYPE(*CHAR) LEN(512)
CHGVAR      VAR(&ADDR) VALUE('INFO@THECOMPAGNY.COM')
CHGVAR      VAR(&SUBJ) VALUE('THE CUSTOMER LIST')
CHGVAR      VAR(&MMSG) VALUE('Hello,%PARA%%PARA%Here +
                             is the list of the customers you +
                             required. The file comes from our AS/400. +
                             Isn''t this great ?%PARA%%PARA%See you +
                             soon !')
/* OPEN THE SESSION */
```

```

CALL          PGM(LNCOPEN)  PARM(&HANDLE &SVRADDR &CCSID)

MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* CONNECT TO THE SERVER ...*/

/* MESSAGE IS SENT TO RECIPIENT &ADDR */

/* SUBJECT IS &SUBJ AND MESSAGE IS &MESG */

/* */

CHGVAR       VAR(&CMD) VALUE('MAILPREP')

CHGVAR       VAR(&PARM1) VALUE(&ADDR)

CHGVAR       VAR(&PARM2) VALUE(&SUBJ)

CALL         PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)

IF           COND(%SST(&RESULT 1 5) =
SNDMSG       MSG(&RESULT) TOUSR(QPGMR)

SUI:
/* TRANSFER THE SP_CUST FILE AND ATTACH IT TO THE MAIL */
CHGVAR       VAR(&CMD) VALUE('DBFXFER')
CHGVAR       VAR(&PARM1) VALUE('%LNCDIR%\SAMPLES\SP_CUST.T
XT')

CHGVAR       VAR(&PARM2) VALUE('LAUNCHER/SP_CUST')
CALL         PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* ATTACH THE FILE */
CHGVAR       VAR(&CMD) VALUE('MAILATT')
CHGVAR       VAR(&PARM1) VALUE('%LNCDIR%\SAMPLES\SP_CUST.T
XT')

CHGVAR       VAR(&PARM2) VALUE(' ')
CALL         PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* SET THE MESSAGE BODY */
CHGVAR       VAR(&CMD) VALUE('MAILTEXT')
CHGVAR       VAR(&PARM1) VALUE(&MESG)
CHGVAR       VAR(&PARM2) VALUE(' ')
CALL         PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* SET THE PRIORITY */
CHGVAR       VAR(&CMD) VALUE('MAILPRTY')
CHGVAR       VAR(&PARM1) VALUE('*HIGH')
CHGVAR       VAR(&PARM2) VALUE(' ')
CALL         PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
MONMSG       MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

```

```

/* SEND THE MESSAGE */
    CHGVAR      VAR(&CMD) VALUE('MAILSEND')
    CHGVAR      VAR(&PARM1) VALUE(' ')
    CHGVAR      VAR(&PARM2) VALUE(' ')
    CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
    MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
    CHGVAR      VAR(&CMD) VALUE('MAILEND')
    CHGVAR      VAR(&PARM1) VALUE(' ')
    CHGVAR      VAR(&PARM2) VALUE(' ')
    CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
                                &PARM2 &RESULT)
    MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

CLOSE:
/* END OF COMMUNICATION WITH THE SERVER STATION ... */
    CHGVAR      VAR(&CMD) VALUE('END ')
    CHGVAR      VAR(&PARM1) VALUE(' ')
    CHGVAR      VAR(&PARM2) VALUE(' ')
    CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                                &PARM1 &PARM2 &RESULT)
    MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

    GOTO END

ERROR:
    CALL        PGM(LNCCLOSE) PARM(&HANDLE)
    RCVMSG      MSG(&MSG1) SECLVL(&MSG2) MSGID(&MSGID)

    SNDPGMMSG   MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                MSGDTA('ERROR !!!!! ' *CAT &MSGID *CAT ' ' +
                ' *CAT &MSG1 *CAT ' ' *CAT &MSG2) +
                MSGTYPE(*ESCAPE)
    GOTO CLOSE

END:
    ENDPGM

```

Exemple W_DEMO

Cet exemple illustre quelques fonctionnalités de Launcher avec Word :

- récupération d'informations sur l'AS/400 envoyées directement dans le document Word (nom de l'utilisateur, numéro de série, l'heure système),
- mise en forme du tableau (bordure)
- insertion d'un document, d'une image...

Voici le résultat avec les éléments transférés



LAUNCHER OFFICE

Numéro de série de l'AS/400 : **4486D0A**

Utilisateur connecté : **QPGMR**

Heure Système : **17:51:04**

Le présent document a été composé par votre AS/400, qui a pu utiliser les fonctionnalités de Word, grâce à LAUNCHER Office.

Le programme que vous venez de lancer a composé le présent document à partir du modèle « LncModel.doc », et des données lues ou calculées sur l'AS/400.

Vous pouvez consulter les sources CL dans le fichier « QCLSRC » de la librairie « LAUNCHER ».

Vous trouverez le modèle « LncModel.doc », dans le sous répertoire « Samples » de LAUNCHER Office sur votre ordinateur.

Le chemin par défaut est :

« \Program files\LAUNCHER<X>\Samples\LncModel.doc »

Insertion de données par l'AS/400.

Des données calculées par le programme AS/400 ont été insérées dans ce document.

Ainsi, le numéro de série, le nom de l'utilisateur connecté, et l'heure système affichés en haut de ce document, sont issus de votre AS/400.

Pour faire ces insertions de valeurs, le programme CL a utilisé des signets de Word. Ce sont des emplacements dans le texte identifiés par un nom.

Pour voir l'emplacement de ces signets, configurez votre Word :

Menu « Outils » « Options », Onglet « Affichage », cochez la case « Signets ».

Pour connaître leur noms : Menu « Insertion » « Signet ».

Insertion de paragraphes externes.

LAUNCHER Office vous permet d'insérer des objets externes dans votre document Word.

Le présent paragraphe n'existe pas dans le document original « LncModel.doc ».

Il existe sous la forme d'un document isolé, sous le nom « LncPara01.doc », dans le répertoire des exemples de LAUNCHER Office.

Le programme AS/400 a demandé à Word d'insérer ce paragraphe.

Cette fonctionnalité pourra vous permettre de construire des documents à partir de clauses, vers lesquelles le programme AS/400 a les références dans sa base de données.

Ce paragraphe inséré pourrait aussi être un texte libre saisi par l'utilisateur, sous Word.

LAUNCHER Office peut demander à l'utilisateur de saisir ou modifier un texte, il peut le sauvegarder dans un fichier isolé, pour ensuite l'insérer dans un autre document à imprimer.

Gestion de tableaux Word.

LAUNCHER Office peut créer et alimenter des tableaux.

Le tableau suivant existe dans le modèle, mais sans aucune autre valeur que les entêtes de colonnes.

Nous avons choisi de lister dans ce tableau les librairies en ligne sur votre AS/400.

Dans une lettre de relance, ce tableau pourrait lister les factures impayées de votre client.

| Librairie | Description | Propriétaire |
|-----------|-------------------------------|--------------|
| LAUNCHER | | QSECOFR |
| QGPL | | QSYS |
| QSYS | Bibliothèque système | QSYS |
| QSYS2 | Bibliothèque système pour CPI | QSYS |

| | | |
|---------|--|------|
| QHLPSYS | | QSYS |
| QUSRSYS | | QSYS |

Insertion d'images.

L'image en haut et à gauche de la première page a été insérée par le programme CL.

Comme pour du texte, un signet est placé dans le modèle.

Le programme CL désigne le fichier « LNCimg01.gif » comme devant être inséré à l'emplacement de ce signet.

et le source CL de l'exemple

```

PGM          PARM(&SVRADDR)
DCL          VAR(&SVRADDR) TYPE(*CHAR) LEN(30)
/* VARIABLES USED TO OPEN COMMUNICATION AND SEND COMMANDS ... */
DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&CCSID) TYPE(*CHAR) LEN(10)
DCL          VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL          VAR(&PARAM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&PARAM2) TYPE(*CHAR) LEN(1024)
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)

DCL          VAR(&MSGID) TYPE(*CHAR) LEN(7)
DCL          VAR(&MSG1) TYPE(*CHAR) LEN(80)
DCL          VAR(&MSG2) TYPE(*CHAR) LEN(80)
DCL          VAR(&USER) TYPE(*CHAR) LEN(10)
DCL          VAR(&SRLN) TYPE(*CHAR) LEN(8)
DCL          VAR(&TIME) TYPE(*CHAR) LEN(7)
DCL          VAR(&ULIBL) TYPE(*CHAR) LEN(275)
DCL          VAR(&SLIBL) TYPE(*CHAR) LEN(165)
DCL          VAR(&LTEXT) TYPE(*CHAR) LEN(50)
DCL          VAR(&LOWNR) TYPE(*CHAR) LEN(10)
DCL          VAR(&LLIB) TYPE(*CHAR) LEN(10)

DCL          VAR(&IDX) TYPE(*DEC) LEN(4)

DCL          VAR(&DOTTED) TYPE(*CHAR) LEN(30)
DCL          VAR(&ROW) TYPE(*DEC) LEN(2)
DCL          VAR(&COL) TYPE(*DEC) LEN(2)
DCL          VAR(&C_ROW) TYPE(*CHAR) LEN(2)
DCL          VAR(&C_COL) TYPE(*CHAR) LEN(2)

RTVJOBA     USER(&USER) USRLIBL(&ULIBL) SYSLIBL(&SLIBL)

RTVSYSVAL   SYSVAL(QSRLNBR) RTNVAR(&SRLN)
RTVSYSVAL   SYSVAL(QTIME) RTNVAR(&TIME)

/* OPEN ONLY ONE CONVERSATION WITH THE SERVER */
CHGVAR     VAR(&HANDLE) VALUE('*ONLY')
```

```

/* TRANSLATE CCSID ACCORDING TO THE JOB CCSID */
      CHGVAR      VAR(&CCSID) VALUE('*JOB')

/* CONNECT TO THE SERVER ...*/
      CALL        PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)

      MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* START WORD ON THE WINDOWS STATION */

      CHGVAR      VAR(&CMD) VALUE('WORDOPEN ')
      CHGVAR      VAR(&PARM1) VALUE(' ')
      CHGVAR      VAR(&PARM2) VALUE(' ')
      CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
      MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

      CHGVAR      VAR(&CMD) VALUE('WORDHIDE')
      CHGVAR      VAR(&PARM1) VALUE(' ')
      CHGVAR      VAR(&PARM2) VALUE(' ')
/*      CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +      */
&PARM1 &PARM2 &RESULT)      */
/*      MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))      */

/* OPEN A DOCUMENT FILE ON THE WINDOWS STATION */

      CHGVAR      VAR(&CMD) VALUE('WNEWFILE ')
      CHGVAR      VAR(&PARM1) VALUE('%LNCDIR%\SAMPLES\LNCCMODEL
.DOC ')
      CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
      MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* WRITE TEXT AT A BOOKMARK */

      CHGVAR      VAR(&CMD) VALUE('WBOOKMARK ')
      CHGVAR      VAR(&PARM1) VALUE('USER')
      CHGVAR      VAR(&PARM2) VALUE(&USER)
      CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)

      CHGVAR      VAR(&PARM1) VALUE('SNAS/400')
      CHGVAR      VAR(&PARM2) VALUE(&SRLN)
      CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)

      CHGVAR      VAR(&PARM1) VALUE('TIME')
      CHGVAR      VAR(&PARM2) VALUE('%SST(&TIME 1 2) *CAT ':' +
*CAT %SST(&TIME 3 2) *CAT ':' *CAT +
%SST(&TIME 5 2))
      CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)

      CHGVAR      VAR(&CMD) VALUE('WBOOKMARK ')

```

```

FRST_ROW:   CHGVAR   VAR(&PARM1) VALUE('TABLE')
NEXT_ROW:   CHGVAR   VAR(&IDX)  VALUE(1)

CHGVAR     VAR(&LLIB) VALUE(%SST(&ULIBL &IDX 10))

CHGVAR     VAR(&IDX) VALUE(&IDX + 11)
IF         COND(&LLIB *EQ '          ') THEN(GOTO +
CMDLBL(END_LIST))
RTVOBJD   OBJ(&LLIB) OBJTYPE(*LIB) TEXT(&LTEXT) +
OWNER(&LOWNR)
MONMSG    MSGID(CPF0000) EXEC(GOTO CMDLBL(NEXT_ROW))

CHGVAR     VAR(&PARM2) VALUE(&LLIB)
CALL      PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
CHGVAR     VAR(&PARM1) VALUE('*RIGHT')
CHGVAR     VAR(&PARM2) VALUE(&LTEXT)
CALL      PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
CHGVAR     VAR(&PARM2) VALUE(&LOWNR)
CALL      PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)

GOTO      CMDLBL(NEXT_ROW)

END_LIST:   CHGVAR     VAR(&ULIBL) VALUE(&SLIBL)
CHGVAR     VAR(&SLIBL) VALUE(*BLANK)

IF         COND(&ULIBL *NE *BLANK) THEN(GOTO +
CMDLBL(FRST_ROW))

CHGVAR     VAR(&CMD) VALUE(WSELECT)
CHGVAR     VAR(&PARM1) VALUE(*ROW)
CHGVAR     VAR(&PARM2) VALUE(' ')
CALL      PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)

CHGVAR     VAR(&CMD) VALUE(WSETPROP)
CHGVAR     VAR(&PARM1) +
VALUE('Selection.Borders.Item(wdBorderBotto+
m).LineStyle')
CHGVAR     VAR(&PARM2) VALUE('wdLineStyleSingle')
CALL      PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
MONMSG    MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

CHGVAR     VAR(&CMD) VALUE('WBOOKMARK ')
CHGVAR     VAR(&PARM1) VALUE('PARA01')
CHGVAR     VAR(&PARM2) VALUE(' ')
CALL      PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)

CHGVAR     VAR(&CMD) VALUE(WINSERTF)
CHGVAR     VAR(&PARM1) +

```



```

VALUE ('%LNCDIR%\SAMPLES\LNCpara01.DOC ')

CALL      PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
MONMSG    MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

CHGVAR    VAR(&CMD) VALUE('WBOOKMARK ')
CHGVAR    VAR(&PARM1) VALUE('IMG01')
CHGVAR    VAR(&PARM2) VALUE(' ')
CALL      PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
CHGVAR    VAR(&CMD) VALUE(WINSERTIMG)
CHGVAR    VAR(&PARM1) +
VALUE('%LNCDIR%\SAMPLES\LNCIMG01.GIF ')

CALL      PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
MONMSG    MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

CHGVAR    VAR(&CMD) VALUE('WORDSHOW')
CHGVAR    VAR(&PARM1) VALUE(' ')
CHGVAR    VAR(&PARM2) VALUE(' ')
CALL      PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
MONMSG    MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

CHGVAR    VAR(&CMD) VALUE('WORDWAIT')
CHGVAR    VAR(&PARM1) VALUE(' ')
CHGVAR    VAR(&PARM2) VALUE(' ')
CALL      PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
MONMSG    MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* END OF COMMUNICATION WITH THE SERVER STATION ... */
CLOSE:
CHGVAR    VAR(&CMD) VALUE('END ')

CHGVAR    VAR(&PARM1) VALUE(' ')
CHGVAR    VAR(&PARM2) VALUE(' ')
/*
CALL      PGM(LNCCMD)  PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
*/
/*
MONMSG    MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
*/
CALL      PGM(LNCCLOSE) PARM(&HANDLE)
MONMSG    MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

```



```

                                GOTO END
ERROR:
CHGVAR      VAR(&CMD) VALUE ('END ')
CHGVAR      VAR(&PARM1) VALUE (' ')
CHGVAR      VAR(&PARM2) VALUE (' ')
CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                                &PARM1 &PARM2 &RESULT)
MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR0))

ERROR0:
CALL        PGM(LNCCLOSE) PARM(&HANDLE)

RCVMSG      MSG(&MSG1) SECLVL(&MSG2) MSGID(&MSGID)

SNDPGMMMSG  MSGID(CPF9898) MSGF(QCPFMSG) +
                                MSGDTA('Error !!!!! ' *CAT &MSGID *CAT ' ' +
                                ' *CAT &MSG1 *CAT ' ' *CAT &MSG2) +
                                MSGTYPE(*ESCAPE)

END:
SNDPGMMMSG  MSGID(CPF9898) MSGF(QCPFMSG) +
                                MSGDTA('Document ''result.doc'' saved by +
                                ' *CAT &USER *CAT 'in samples directory') +
                                MSGTYPE(*COMP)

ENDPGM
```

Exemple X_DEMO

Cet exemple présente quelques fonctions de LAUNCHER Office avec Excel.

Gestion de feuilles

On commence par l'insertion d'une nouvelle feuille par la commande XLADDSHEET puis le changement de son nom par la commande XLSHNAME. LAUNCHER Office permet aussi de copier ou de supprimer les feuilles du classeur.

Insertion de données

Parmi les nombreuses commandes pour alimenter la feuille, la plus simple et la plus pratique est XLSETLINE qui écrit en une fois toute la ligne.

Le premier paramètre contient les données, séparées par le mot clé %SEP%.

Le second paramètre peut contrôler la mise en forme de chaque cellule.

Nous écrivons donc une première ligne d'entête avec une mise en gras puis deux lignes de valeurs, donc la deuxième est formatée avec 2 décimales par le mot clé NUMFMT(2).

Création d'un graphique

Une fois le tableau rempli la commande XLDRAWGR nous permet de créer le graphique correspondant sur une nouvelle feuille. Dans ce cas un graphique de type "Aires" mais on peut évidemment utiliser tous les types de graphiques disponibles.

Nous modifions ensuite certaines propriétés du graphique ; titre et nom des séries par la commande XLSETPROP.

Une fois les opérations effectuées on peut rendre Excel visible par la commande EXCELSHOW.

Source de l'exemple

```
PGM
DCL          VAR(&SVRADDR) TYPE(*CHAR) LEN(30)

/* DECLARATION OF VARIABLES NEEDED FOR A CONNECTION TO THE SERVER  *
/

DCL          VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL          VAR(&CCSID) TYPE(*CHAR) LEN(10)
DCL          VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL          VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL          VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL          VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL          VAR(&RESULT) TYPE(*CHAR) LEN(512)
DCL          VAR(&MSGID) TYPE(*CHAR) LEN(7)
DCL          VAR(&MSG1) TYPE(*CHAR) LEN(80)
```

```

                DCL          VAR(&MSG2) TYPE(*CHAR) LEN(80)
                CHGVAR      VAR(&SVRADDR) VALUE('*DEV')
/* OPEN ONLY ONE CONVERSATION WITH THE SERVER */
                CHGVAR      VAR(&HANDLE) VALUE('*ONLY')
/* TRANSLATE THE CCSID ACCORDING TO THE JOB CCSOD */
                CHGVAR      VAR(&CCSID) VALUE('*JOB')
/* CONNECTION WITH THE SERVER */
                CALL        PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* OPENING EXCEL */
                CHGVAR      VAR(&CMD) VALUE('EXCELOPEN ')
                CHGVAR      VAR(&PARM1) VALUE(' ')
                CHGVAR      VAR(&PARM2) VALUE(' ')
                CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* ADD A NEW SHEET */
                CHGVAR      VAR(&CMD) VALUE('XLADDSHEET ')
                CHGVAR      VAR(&PARM1) VALUE(' ')
                CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* RENAME THE SHEET */
                CHGVAR      VAR(&CMD) VALUE('XLSHNAME ')
                CHGVAR      VAR(&PARM1) VALUE('VENTES 1ER TRIMESTRE')

                CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))

/* INSERT FIRST LINE WITH HEADERS */
                CHGVAR      VAR(&CMD) VALUE('XLSETLINE ')
                CHGVAR      VAR(&PARM1) VALUE('JANVIER%SEP%FEVRIER%SEP%MA
RS')
                CHGVAR      VAR(&PARM2) VALUE('PROP(FONT.BOLD)=
PROP(FONT.BOLD)=')
                CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* FIRST DATA LINE */
                CHGVAR      VAR(&CMD) VALUE('XLSETLINE ')
                CHGVAR      VAR(&PARM1) VALUE('000123%SEP%22250%SEP%301')

                CHGVAR      VAR(&PARM2) VALUE('FORMAT()%SEP%NUMFMT(2)')

                CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* SECOND DATA LINE */
                CHGVAR      VAR(&PARM1) VALUE('223%SEP%23250%SEP%285')

                CHGVAR      VAR(&PARM2) VALUE('FORMAT()%SEP%NUMFMT(2)')

                CALL        PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
                &PARM1 &PARM2 &RESULT)
                MONMSG      MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* INSERT THE GRAPHIC */
                CHGVAR      VAR(&CMD) VALUE('XLDRAWGR ')
                CHGVAR      VAR(&PARM1) VALUE('A1:C3')

```

```

0'' ''+
CHGVAR VAR(&PARM2) VALUE(''AREA'' ''ROWS'' ''1'' ''
MARKET'' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* CHANGING THE GRAPHS TITLE */
CHGVAR VAR(&CMD) VALUE('XLSETPROP')
CHGVAR VAR(&PARM1) VALUE('ACTIVECHART.CHARTTITLE.CHA
RACTERS.+
TEXT')
CHGVAR VAR(&PARM2) VALUE('VENTES TRIMESTRE 1')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* CHANGE FIRST SERIE NAME */
CHGVAR VAR(&CMD) VALUE('XLSETPROP')
CHGVAR VAR(&PARM1) VALUE('ACTIVECHART.SERIESCOLLECTI
ON(1)+
.NAME')
CHGVAR VAR(&PARM2) VALUE('PRODUIT A')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* SECOND SERIE */
CHGVAR VAR(&CMD) VALUE('XLSETPROP')
CHGVAR VAR(&PARM1) VALUE('ACTIVECHART.SERIESCOLLECTI
ON(2)+
.NAME')
CHGVAR VAR(&PARM2) VALUE('PRODUIT B')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* MAKE EXCEL VISIBLE */
CHGVAR VAR(&CMD) VALUE('EXCELSHOW ')
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* END */
CHGVAR VAR(&CMD) VALUE('END ')
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT +
&PARM1 &PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* END OF COMMUNICATION...*/
CALL PGM(LNCCLOSE) PARM(&HANDLE)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
GOTO END
ERROR:
RCVMSG MSG(&MSG1) SECLVL(&MSG2) MSGID(&MSGID)

```



```
SNDPGMMSG MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +  
MSGDTA('ERROR !!!!! ' *CAT &MSGID *CAT ' +  
' *CAT &MSG1 *CAT ' ' *CAT &MSG2) +  
MSGTYPE(*ESCAPE)
```

END:

ENDPGM

Exemples de programmes RPG

Exemple Word

Le principe de cet exemple est d'aller remplir le tableau des commandes.

Pour se faire, nous avons placé un signet à la première cellule de donnée (deuxième ligne, car la première est l'entête du tableau).

Donc, le programme créé un document à partir du modèle, va inscrire les données de l'entête, puis remplit le tableau (à l'aide de la commande [WSETLINE](#)).

Le tableau ne contient qu'une seule ligne de données, parce que Word ajoute automatiquement une ligne suivante lorsque, une fois que vous êtes en fin de ligne et que vous passez à la colonne suivante (par Tab. par exemple).

Ce qui fait que l'on aura pas de ligne vide.

La zone de texte qui suit le tableau se décalera au fur et à mesure que le tableau grandit.

Source du programme RPG pour Word

```
*****
* Fichiers utilisés:   SP_CUST --> Fichier clients          *
*                    SP_ORD_ID --> Fichier commandes (ORDERS) LF*
*                    SP_DET_OR --> Fichier détail (DETAIL) LF  *
*****
      FSP_CUST   if   e           k disk
      FSP_ORD_ID if   e           k disk
      FSP_DET_OR if   e           k disk

*****
* Déclaration des variables utilisées par LAUNCHER Office.      *
* Voir le chapitre 5.                                          *
*****
      DHANDLE           s           50      inz('*ONLY')
      DSVRADDR          s           30      inz('*DEV')
      DCCSID            s           10      inz('*JOB')
      DLNCCMD           s           10      inz(*blanks)
      DLNCPARM1         s           512     inz(*blanks)
      DLNCPARM2         s           1024    inz(*blanks)
      DLNCOPT           s            1      inz(*blanks)
      DLNCRESLT         s           512     inz(*blanks)

*****
* Variables utilisées par le programme                          *
*****
      DPREMIER          s            9      inz(*blanks)
      DDATECMD          s            d      datfmt(*EUR)
      DDATECMDS         s           10      inz(*blanks)
      DPRIXUP           s            8p 0    inz(0)
      DPRIXTP           s            8p 0    inz(0)
      DPRIXU            s            8      inz(*blanks)
      DPRIXT            s            8      inz(*blanks)
      DQTE              s           10      inz(*blanks)
      DRECORD           s            9
```

```

*****
* Point d'entrée du programme *
*****
c      *ENTRY      PLIST
c      premier    PARM                record
*****
* Chaînage sur le client voulu *
*****
c      PREMIER    chain    SP_CUST                90
*
* Connexion à LAUNCHER Office. Le serveur est adressé par la variable
* SVRADDR. *DEV signifie que l'on travaille sur le serveur qui émule
* l'écran courant (utilisation du écran 5250).
*
c      call      'LNCOPEN'
c      parm      handle
c      parm      svraddr
c      parm      ccsid
*-----*
* Edits a sales document that will be sent by e-mail
*-----*
c      eval      lnccmd='WORDOPEN'
c      exsr      CALLLNC
c      eval      lnccmd='WMINIMIZE'
c      exsr      CALLLNC
* Création d'un nouveau document sur la base d'un modèle existant
c      eval      lnccmd='WNEWFILE'
c      eval      lncparm1='%LNCDIR%\Samples' +
c              '\Letter.dot'
c      exsr      CALLLNC
c      eval      lnccmd='WMINIMIZE'
c      exsr      CALLLNC
* A l'aide des signets, les données sont affectées
c      eval      lnccmd='WBOOKMARK'
c      eval      lncparm1='FNAME'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='LNAME'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='ADDRESS'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='ZIP'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='CITY'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='STATE'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='PHONE'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='FAX'
c      eval      lncparm2=
c      exsr      CALLLNC
c      eval      lncparm1='EMAIL'
c      eval      lncparm2=
c      exsr      CALLLNC
* Routine qui va lire toutes les commandes du client.
* Travail sur le fichier de commandes
c      cust_id    setll    sp_ord_id
c      cust_id    reade    sp_ord_id

```



```

c 91 goto Fin
* Déplacement vers la table
c eval lnccmd='WBOOKMARK'
c eval lncparm1='TABLE'
c eval lncparm2=
c exsr CALLLNC
c loop_order tag
c move ORDER_DATE datecmd
c movel datecmd datecmds
* Lecture du détail de la commande
c order_id setll sp_det_or
c order_id reade sp_det_or 92
c 92 goto Suite
c loop_detail tag
c eval lnccmd='WSETLINE'
c eval lncparm1=datecmds + '%SEP%' +
PART_DESC
c exsr CALLLNC
c eval PRIXUP=
c movel prixup prixu
c eval lnccmd='WBOOKMARK'
c eval lncparm1=' '*RIGHT' 'NUMFMT(0)' ' '
c eval lncparm2=
c exsr CALLLNC
c movel(p) QUANTITY QTE
c eval lncparm2=
c exsr CALLLNC
c eval PRIXTP=
c movel prixtp prixt
c eval lncparm2=
c exsr CALLLNC
* Détail suivant
c order_id reade sp_det_or 92
c N92 eval lncparm1='*RIGHT'
c N92 eval lncparm2=
c N92 exsr CALLLNC
c N92 goto loop_detail
c Suite tag
* Commande suivante
c cust_id reade sp_ord_id 93
c N93 eval lncparm1='*RIGHT'
c N93 eval lncparm2=
c N93 exsr CALLLNC
c N93 goto loop_order
* Sauvegarde, et envoi
c Fin tag
c eval lnccmd='WSAVEAS'
c eval lncparm1='%LNCDIR%\samples\' +
firstname + '.doc'
c eval lncparm2=
c exsr CALLLNC
c eval lnccmd='WSENDTO'
c eval lncparm1=
* La ligne suivante n'est pas exécutée
* c exsr CALLLNC
c eval lnccmd='WMAXIMIZE'
c exsr CALLLNC
* Quitte le serveur LAUNCHER Office
c eval lnccmd='END'
c exsr CALLLNC
c call 'LNCCLDSE'
c parm HANDLE
c seton lr

```

* SUB CALLLNC : Cette sous-routine envoie la commande à LAUNCHER Office.
* Les différents paramètres sont déjà affectés.

* Voir le chapitre 5 pour leur utilisation

```

*-----
c      CALLLNC      BEGSR
c      call        'LNCCMD'
c      parm
c      parm        HANDLE
c      parm        LNCCMD
c      parm        LNCOPT
c      parm        LNCPARM1
c      parm        LNCPARM2
c      parm        LNCRESLT
c      ENDSR
  
```

Exemple Excel

Ici, grâce aux noms des cellules, il est aussi facile que Word de remplir les données de l'entête.

Le remplissage du tableau des commandes se fera avec la commande [XLSETLINE](#).

Puis, pour délimiter le tableau, nous faisons appel à une macro.

Source du programme RPG pour Excel

```

*****
* Fichiers utilisés:   SP_CUST --> Fichier clients          *
*                    SP_ORD_ID --> Fichier commandes (ORDERS) LF*
*                    SP_DET_OR --> Fichier détail (DETAIL) LF  *
*****
      FSP_CUST   if   e           k disk
      FSP_ORD_ID if   e           k disk
      FSP_DET_OR if   e           k disk

*****
* Déclaration des variables utilisées par LAUNCHER/400.      *
*****
      DHANDLE      s           50   inz ('*ONLY')
      DSVRADDR     s           30   inz ('*DEV')
      DCCSID       s           10   inz ('*JOB')
      DLNCCMD      s           10   inz (*blanks)
      DLNCPARM1    s           512  inz (*blanks)
      DLNCPARM2    s           1024 inz (*blanks)
      DLNCOPT      s            1   inz (*blanks)
      DLNCRESLT    s           512  inz (*blanks)

*****
* Variables utilisées par le programme                      *
*****
      DPREMIER     s            9   inz (*blanks)
      DDATECMD     s            d   datfmt (*EUR)
      DDATECMDS    s           10   inz (*blanks)
      DPRIXUP      s           8p 0  inz (0)
      DPRIXTP      s           8p 0  inz (0)
      DPRIXU       s            8   inz (*blanks)
      DPRIXT       s            8   inz (*blanks)
      DQTE         s           10   inz (*blanks)
      DRECORD      s            9
  
```



```

*****
* Point d'entrée du programme *
*****
      c      *ENTRY      PLIST
      c      premier    PARM                      record
*****
* Chaînage sur le client voulu *
*****
      c      PREMIER    chain    SP_CUST          90
*
* Connexion à LAUNCHER/400. Le serveur est adressé par la variable
* SVRADDR. *DEV signifie que l'on travaille sur le serveur qui émule
* l'écran courant (utilisation du écran 5250).
*
      c              call    'LNCOPEN'
      c              parm    handle
      c              parm    svraddr
      c              parm    ccsid
*-----*
* Edition d'un document de vente à envoyer par mail
*-----*
      c              eval    lnccmd='WORDOPEN'
      c              exsr    CALLLNC
      c              eval    lnccmd='WMINIMIZE'
      c              exsr    CALLLNC
* Création d'un nouveau document sur la base d'un modèle existant
      c              eval    lnccmd='XLOPENFILE'
      c              eval    lncparm1='%LNCDIR%\Samples'+
      c              '\letter.xlt'
      c              exsr    CALLLNC
      c              eval    lnccmd='XLMINIMIZE'
      c              exsr    CALLLNC
* Utilisation des noms de cellule pour l'entête
      c              eval    lnccmd='XLSETTEXT'
      c              eval    lncparm1='NAME'
      c              eval    lncparm2=firstname + lastname
      c              exsr    CALLLNC
      c              eval    lncparm1='ADDRESS'
      c              eval    lncparm2=
      c              exsr    CALLLNC
      c              eval    lncparm1='ZIP'
      c              eval    lncparm2=
      c              exsr    CALLLNC
      c              eval    lncparm1='CITY'
      c              eval    lncparm2=
      c              exsr    CALLLNC
      c              eval    lncparm1='STATE'
      c              eval    lncparm2=
      c              exsr    CALLLNC
      c              eval    lncparm1='PHONE'
      c              eval    lncparm2=
      c              exsr    CALLLNC
      c              eval    lncparm1='FAX'
      c              eval    lncparm2=
      c              exsr    CALLLNC
      c              eval    lncparm1='EMAIL'
      c              eval    lncparm2=
      c              exsr    CALLLNC
* Routine qui va lire toutes les commandes du client.

```

```

* Travail sur le fichier de commandes
  c      cust_id      setll      sp_ord_id
  c      cust_id      reade      sp_ord_id      91
  c      91           goto      Fin
* On va à la première cellule de donnée
  c      eval          lnccmd='XLGOTOCELL'
  c      eval          lncparm1='TABLE'
  c      eval          lncparm2=
  c      exsr          CALLLNC
  c      loop_order   tag
  c      move          ORDER_DATE      datecmd
  c      movel         datecmd         datecmds
* Puis le détail de chaque commande
  c      order_id     setll      sp_det_or
  c      order_id     reade      sp_det_or      92
  c      92           goto      Suite
  c      loop_detail  tag
  c      eval          lnccmd='XLSETLINE'
  c      eval          PRIXUP=
  c      movel         prixup          prixu
  c      movel (p)    QUANTITY          QTE
  c      eval          PRIXTP=
  c      movel         prixtp          prixt
  c      eval          lncparm1=datecmds + '%SEP%' +
  c      PART_DESC + '%SEP%' + prixu +
  c      '%SEP%' + QTE + '%SEP%'
  c      + PRIXT
  c      exsr          CALLLNC
* On lit le prochain détail de commande
  c      order_id     reade      sp_det_or      92
  c      N92          goto      loop_detail
  c      Suite        tag
* On lit ici la prochaine commande
  c      cust_id      reade      sp_ord_id      93
  c      N93          goto      loop_order
* Sauvegarde et envoi
  c      Fin          tag
* Petite macro pour faire la mise en page
  c      eval          lnccmd='XLEXEMACRO'
  c      eval          lncparm1='PageSetup'
  c      exsr          CALLLNC
  c      eval          lnccmd='XLSAVEAS'
  c      eval          lncparm1='%LNCDIR%\samples\'
  c      + firstname + '.xls'
  c      eval          lncparm2=
  c      exsr          CALLLNC
  c      eval          lnccmd='XLSENDTO'
  c      eval          lncparm1=
* La ligne suivante n'est pas exécutée
  *   c      exsr          CALLLNC
  c      eval          lnccmd='XLMAXIMIZE'
  c      exsr          CALLLNC
  c      seton          lr
* Déconnexion du serveur
  c      eval          lnccmd='END'
  c      exsr          CALLLNC
  c      call          'LNCCLOSE'
  c      parm          HANDLE

```

FAQ

*TCAT et *CAT et *BCAT

Ces opérateurs permettent de concaténer deux chaînes de caractères.

Exemples

* **CAT** : Concatène deux chaînes de caractères.

Les blancs en fin de la première chaîne de caractère et en début de la deuxième chaîne ne sont pas tronqués.

Par exemple :

'ABC' *CAT 'DEF' devient 'ABCDEF'

'ABC ' *CAT 'DEF ' devient 'ABC DEF '

* **BCAT** : Tronque tous les blancs à la fin de la première chaîne de caractères.

Un blanc est inséré entre les deux chaînes de caractères.

Les blancs en début de la deuxième chaîne ne sont pas tronqués.

Par exemple :

'ABC' *BCAT 'DEF' devient 'ABC DEF'

'ABC ' *BCAT 'DEF' devient 'ABC DEF'

* **TCAT** : Tronque tous les blancs à la fin de la première chaîne de caractères.

Les blancs en début de la deuxième chaîne ne sont pas tronqués.

Par exemple :

'ABC' *TCAT 'DEF' devient 'ABCDEF'

'ABC ' *TCAT 'DEF' devient 'ABCDEF'

'ABC' *TCAT ' DEF' devient 'ABC DEF'

'ABC ' *TCAT ' DEF' devient 'ABC DEF'

Afficher le groupe de processeur AS/400 - Pour information

Pour connaître le groupe de processeur de son AS/400 quand EASYCOM ou LAUNCHER ne sont pas installés :

- Sur une session Terminal sous le profile **QSECOFR**, tapez la commande suivante :

WRKLCINF <Entrée>

L'écran suivant s'affiche, avec en 4^{ème} ligne le groupe de processeur :

```
WORK WITH LICENSE INFORMATION S1234567
20/01/19 16:22:21
SYSTEM SERIAL NUMBER . . . . . : 1234567
PROCESSOR GROUP . . . . . : P05
```

Quelques commandes CL

| | |
|--------------------------------|-----------|
| Appeler un programme | CALL |
| Modifier une variable | CHGVAR |
| Copyright CALL | COPYRIGHT |
| Créer un programme CL lié | CRTBNDCL |
| Créer un module CL | CRTCLMOD |
| Créer un programme CL | CRTCLPGM |
| Convertir un source CL 38 | CVTCLSRC |
| Convert TCP/IP CL Source | CVTTCPCL |
| Convertir le format de date | CVTDAT |
| Données | DATA |
| Déclarer une variable CL | DCL |
| Début de groupe DO | DO |
| Appeler un programme | CALL |
| Modifier une variable | CHGVAR |
| Copyright | COPYRIGHT |
| Créer un programme CL lié | CRTBNDCL |
| Créer un module CL | CRTCLMOD |
| Créer un programme CL | CRTCLPGM |
| Convertir un source CL 38 | CVTCLSRC |
| Convert TCP/IP CL Source | CVTTCPCL |
| Convertir le format de date | CVTDAT |
| Données | DATA |
| Déclarer une variable CL | DCL |
| Déclarer un fichier | DCLF |
| Supprimer un module | DLTMOD |
| Supprimer un programme | DLTPGM |
| Clicher le programme CL | DMPCLPGM |
| Début de groupe DO | DO |
| Afficher un module | DSPMOD |
| Afficher un programme | DSPPGM |
| Afficher références programmes | DSPPGMREF |
| Sinon | ELSE |
| Fin de groupe DO | ENDDO |

| | |
|--------------------------------|------------|
| Fin des données | ENDINP |
| Fin de programme | ENDPGM |
| Arrêter la réception | ENDRCV |
| Aller à | GOTO |
| Si | IF |
| Intercepter message | MONMSG |
| Programme | PGM |
| Définition de qualificatif | QUAL |
| Recevoir fichier écran ou BD | RCVF |
| Recevoir un message | RCVMSG |
| Retour | RETURN |
| Enlever un message | RMVMSG |
| Extraire un source CL | RTVCLSRC |
| Extraire une zone de données | RTVDTAARA |
| Extraire un poste de journal | RTVJRNE |
| Extraire description de membre | RTVMBRD |
| Extraire un message | RTVMSG |
| Envoyer fichier écran | SNDF |
| Envoyer un message programme | SNDPGMMMSG |
| Envoyer/recevoir fichier écran | SNDRCVF |
| Envoyer une réponse | SNDRPY |
| Envoyer un message utilisateur | SNDUSRMSG |
| Démarrer SEU | STRSEU |
| Transférer contrôle à pgm | TFRCTL |
| Attendre | WAIT |

Comment créer un fichier de fusion

La commande **DBXFER** permet de transférer directement un fichier de l'AS/400 vers le PC avec différentes options de formatage (**DBXPROP**).

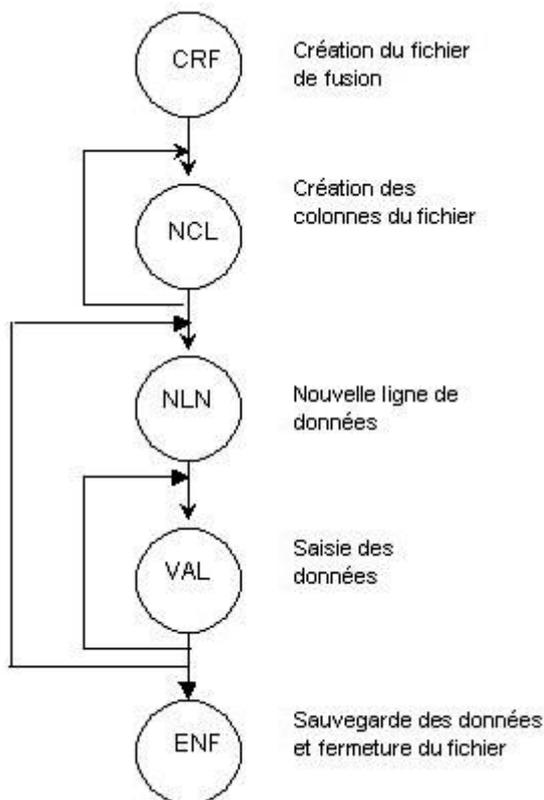
Dans la majorité des cas elle est plus simple et rapide que la procédure de création personnalisée qui suit.

LAUNCHER Office permet de créer un fichier PC rempli de données provenant de l'AS/400. Le fichier sera au format ASCII délimité. Il pourra ainsi être traité par une quelconque application.

La commande **CRF** permet la création du fichier de données.

L'ajout de colonnes dans le fichier de données se fait par la commande **NCL**.

L'ajout de lignes se fait par un appel à **NLN**.



L'ajout de valeurs, dans une colonne donnée, se fait par la commande **VAL**. L'entrée de la valeur se fera toujours dans la ligne courante. Créer une nouvelle ligne condamne la saisie dans la ligne précédente.

La sauvegarde des données et la clôture du fichier se font par un appel à la commande **ENF**.

Il est également possible de réaliser cette opération en une seule fois en réalisant un transfert de données par la commande **DBXFER**. Grâce à cette commande, les données sont exportées au format CSV, directement exploitable par Excel.

Comment modifier la taille des paramètres PARM1 et PARM2 ?

Voir : [Commande PROPERTY](#)

Comment envoyer un spoule dans Word ?

Avec LAUNCHER Office, il est possible de récupérer le contenu d'un fichier spoule dans un document Word.

1° Il vous faut tout d'abord créer et compiler un fichier physique avec la description suivante :

Un seul champ est nécessaire.
Ce champ doit contenir minimum 236 caractères.

2° Une fois compilé, en PCFILE par exemple, vous pouvez utiliser la commande suivante pour copier le fichier spoule dans le fichier physique :

```
CPYSPLF FILE(SPLFNAME) TOFILE(BIBLIO/PCFILE) MBROPT(*REPLACE)
```

3° On peut alors transférer le fichier sur le PC et l'ouvrir dans Word.

Exemple

```
PGM
/* DECLARATIONS */
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50)
DCL VAR(&CMD) TYPE(*CHAR) LEN(10)
DCL VAR(&OPT) TYPE(*CHAR) LEN(1)
DCL VAR(&PARM1) TYPE(*CHAR) LEN(512)
DCL VAR(&PARM2) TYPE(*CHAR) LEN(1024)
DCL VAR(&RESULT) TYPE(*CHAR) LEN(512)
/* OUVERTURE DE LAUNCHER */
LNCOPEN
/* TRANSFERT DU FICHIER SPOULE */
CHGVAR VAR(&CMD) VALUE('DBFXFER')
CHGVAR VAR(&PARM1) VALUE('C:\TEMP\SPOULE.TXT')
CHGVAR VAR(&PARM2) VALUE('MAGALIE/PCFILER')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* OUVERTURE DE WORD */
CHGVAR VAR(&CMD) VALUE('WORDOPEN')
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* WORD VISIBLE */
CHGVAR VAR(&CMD) VALUE('WORDSHOW')
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* OUVERTURE DU FICHIER */
CHGVAR VAR(&CMD) VALUE('WOPENFILE')
```

```
CHGVAR VAR(&PARM1) VALUE('C:\TEMP\SPOULE.TXT')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
/* FERMETURE DE WORD */
CHGVAR VAR(&CMD) VALUE('WORDCLOSE')
CHGVAR VAR(&PARM1) VALUE(' ')
CHGVAR VAR(&PARM2) VALUE(' ')
CALL PGM(LNCCMD) PARM(&HANDLE &CMD &OPT &PARM1 +
&PARM2 &RESULT)
MONMSG MSGID(LNC0000) EXEC(GOTO CMDLBL(ERROR))
LNCCLOSE
ENDPGM
```

Comment insérer du texte dans l'entête ou le pied de page ?

La solution la plus simple consiste à **mettre un signet directement dans l'entête ou le pied**, il sera alors possible d'insérer le texte par la commande WBOOKMARK.

On peut également utiliser la propriété d'affichage SeekView avec la constante WdSeekCurrentPageHeader pour se positionner dans l'entête ou WdSeekCurrentPageFooter pour aller dans le pied puis insérer directement le texte par WTYPETEXT ou WCOPY.

Exemple :

```
PGM
LNCOPEN
LNCCMD CMD(WORDOPEN)
LNCCMD CMD(WOPENFILE) +
PARM1 ('%LNCDIR%\SAMPLES\MODELE_STYLE.DOC')
LNCCMD CMD(WSETPROP) +
PARM1 (activewindow.activepane.view.seekview+
) PARM2 (wdseekcurrentpageheader)
LNCCMD CMD(WTYPETEXT) PARM1 ('CECI EST UNE ENTETE DE PAGE')
LNCCMD CMD(WSETPROP) +
PARM1 (ACTIVIEWINDOW.ACTIVEPANE.VIEW.SEEKVIEW+
) PARM2 (WDSEEKCURRENTPAGEFOOTER)
LNCCMD CMD(WTYPETEXT) PARM1 ('CECI EST UN PIED DE PAGE')
LNCCMD CMD(WORDSHOW)
LNCCLOSE
ENDPGM
```

Autre Exemple :

```
PGM
LNCOPEN
LNCCMD CMD(WORDOPEN)
LNCCMD CMD(WOPENFILE) +
PARM1 ('%LNCDIR%\SAMPLES\MODELE_STYLE.DOC')
LNCCMD CMD(WBOOKMARK) PARM1(entetePAGE) PARM2('Ceci est +
une entête de page')
LNCCMD CMD(WBOOKMARK) PARM1(piedpage) PARM2('Ceci +
est un pied de page')
LNCCMD CMD(WORDSHOW)
LNCCLOSE
ENDPGM
```

Peut-on utiliser Launcher Office sur un MAC ?

Non, actuellement Launcher Office ne peut tourner que sur une machine Windows.



Peut-on lancer un programme dans un batch ?

Oui, cependant il faut installer Launcher Office sur le PC cible.

Attention : Pensez à désactiver les veilles.

Voir : [LNCOPEN - Programme API](#)

Récupérer l'adresse IP du PC

Existe-t-il une commande permettant de récupérer le nom du profil utilisateur PC qui émule une session 5250 ?

Le programme CL suivant appelle LNCOPEN pour recevoir l'adresse IP du PC en cours d'émulation 5250 pour le job courant.

En entrée :

```
&HANDLE = '*GETDOT'
```

```
&SVRADDR = '*DEV'
```

En sortie :

```
&SVRADDR = adresse IP du PC
```

```
PGM
```

```
/* VARIABLES USED TO OPEN COMMUNICATIN AND SEND COMMANDS ...*/
```

```
DCL VAR(&SVRADDR) TYPE(*CHAR) LEN(30)
```

```
DCL VAR(&HANDLE) TYPE(*CHAR) LEN(50)
```

```
DCL VAR(&CCSID) TYPE(*CHAR) LEN(10)
```

```
/* CALL LNCOPEN WITH &HANDLE=*GETDOT, &SVRADDR=*DEV */
```

```
CHGVAR VAR(&SVRADDR) VALUE(*DEV)
```

```
CHGVAR VAR(&HANDLE) VALUE(*GETDOT)
```

```
CHGVAR VAR(&CCSID) VALUE(*JOB)
```

```
CALL PGM(LNCOPEN) PARM(&HANDLE &SVRADDR &CCSID)
```

```
/* ON RETURN, &SVRADDR=IP ADDRESS */
```

```
END:
```

```
SNDPGMMSG MSGID(CPF9898) MSGF(QCPFMSG) +
```

```
MSGDTA(&SVRADDR) MSGTYPE(*COMP)
```

```
ENDPGM
```

Erreur en Mode Service lors de l'utilisation de Word, Excel, Access

PROBLEME: en mode 'service', je n'arrive pas à utiliser Word, Excel, Access ou une application OLE quelconque. J'obtiens le message 'Word n'a pu être lancé', ou 'Excel n'a pu être lancé'.

Attention: cette solution n'est utile que si vous utilisez LAUNCHER Office en **mode service NT**.

Si le service LAUNCHER Office est configuré pour tourner **sous un profil utilisateur précis**, pour pouvoir utiliser Word, Excel et Access il est nécessaire que ce profil ait les autorisations requises.

Pour accorder ces autorisations, deux solutions existent:

Accorder les droits d'administration sur la machine LAUNCHER Office

Cette solution est la plus simple à mettre en œuvre. Elle consiste à rajouter ce profil dans le groupe 'Administrateurs' de la machine où s'exécute le Service LAUNCHER Office.

Rajouter les droits d'exécution COM des applications Excel, Word et Access.

Suivre les étapes suivantes, pour chaque application (Word, Excel, et Access) :

- a. Lancer l'application (Word, Excel ou Access), sous l'utilisateur en question et ouvrir le gestionnaire de macros (par ALT-F11).
- b. La refermer, ainsi que toute application Office.
- c. Lancer 'dcomcnfg' par le menu 'Démarrer/Exécuter'
- d. Sur l'onglet principal, Sélectionner l'application à modifier.

Pour Word, sélectionner 'Document Microsoft Word'.

Pour Excel, sélectionner 'Application Microsoft Excel'.

Pour Access, sélectionner 'Application Microsoft Access'.

Cliquer sur Propriétés, puis sélectionner l'onglet 'Sécurité'.

Cliquer ensuite sur 'Utiliser des autorisations d'exécution personnalisées'.

Puis cliquer sur le bouton 'Modifier' correspondant à ce choix.

Ici, on voit bien qu'un utilisateur 'normal' n'a pas l'autorisation de lancement.

Cliquer sur 'Ajouter', puis sélectionner l'utilisateur ou le groupe désiré.

Recommencer éventuellement toute l'opération pour une autre application (Word, Excel Access, ou autre).

Comment imprimer un document .PDF sans l'ouvrir sous Word ?

1. Transformez le document Word en document PDF
2. Imprimez le document .PDF sans l'ouvrir sous Word
LNCCMD CMD(SHELL) PARM1('c:\temp\test.pdf')
PARM2('visible=false;wait=false;action=print')

Voir : [Commande SHELL](#)

Note : La commande SHELL avec l'option 'action=print' ne permet pas de choisir l'imprimante que l'on veut utiliser. L'impression se fait sur l'imprimante définie par défaut.

Comment supprimer une feuille Excel ?

Quand on veut supprimer une feuille Excel avec l'instruction XLDELSHEET, le système envoie une fenêtre pour confirmation.

Comment faire pour avoir la réponse OK par défaut ?

Une solution pour éviter quelques fenêtres de Excel, est de fixer la propriété "DisplayAlerts" à "False".

Exemple:

Utiliser LNCCMD avec la commande **XLSETPROP** :

```
CHGVAR VAR(&CMD) VALUE(XLSETPROP)
CHGVAR VAR(&PARM1) VALUE('displayalerts')
CHGVAR VAR(PARM2) VALUE(false)
```

Comment mettre à jour un tableau dynamique ?

La syntaxe doit être similaire à :

```
LNCCMD CMD(XLMETHOD) PARM1('ActiveSheet.PivotTables("<nom du tableau croisé dynamique>").PivotCache().Refresh')
```

Ou bien :

```
LNCCMD CMD(XLMETHOD) PARM1('Sheets.Item("<nom de la feuille contenant le tableau>").PivotTables("<nom du tableau croisé dynamique>").PivotCache().Refresh')
```

Envoi de fax avec LAUNCHER Office

Si vous avez un système de fax sur votre PC ou sur votre réseau, alors vous devriez pouvoir l'utiliser de votre AS/400 avec LAUNCHER Office.

Le système de fax system peut être un "Printer Fax driver" ou un "Domino/Exchange fax connector".

1) Printer Fax driver

Avec un tel système, pour envoyer un document par fax, vous devez simplement l'imprimer, en utilisant un driver d'imprimante configuré sur votre PC.

Alors le driver de l'imprimante vous demande le numéro de téléphone ; ou le numéro de téléphone est inclus dans le document, cache, avec un préfixe spécial.

Pour LAUNCHER Office, vous devez pouvoir inclure le numéro de téléphone dans le document.

Consultez la documentation de votre logiciel de fax pour ce dispositif.

Dans le document, le numéro de téléphone doit être inséré par l'intermédiaire d'un champ de fusion, ou par l'intermédiaire d'un signet.

Utilisez la commande LNCPRDOC pour fusionner le document, et la base de données contenant le numéro de téléphone et tous les autres champs de fusion.

Utilisez la commande LNCCMD pour l'impression :

```
LNCCMD CMD(WPRINT) PARM1('Printer="Name of fax/printer driver"')
```

2) Domino or Exchange fax connector

Avec un connector on Domino or Exchange, pour envoyer un fax, vous devez créer un nouveau message, documents attachés.

Le numéro de téléphone est donné au destinataire avec un préfixe spécial.

Exemple : FAX:555 555 555

Avec LAUNCHER Office, vous pouvez envoyer des fax comme vous envoyez des mails.

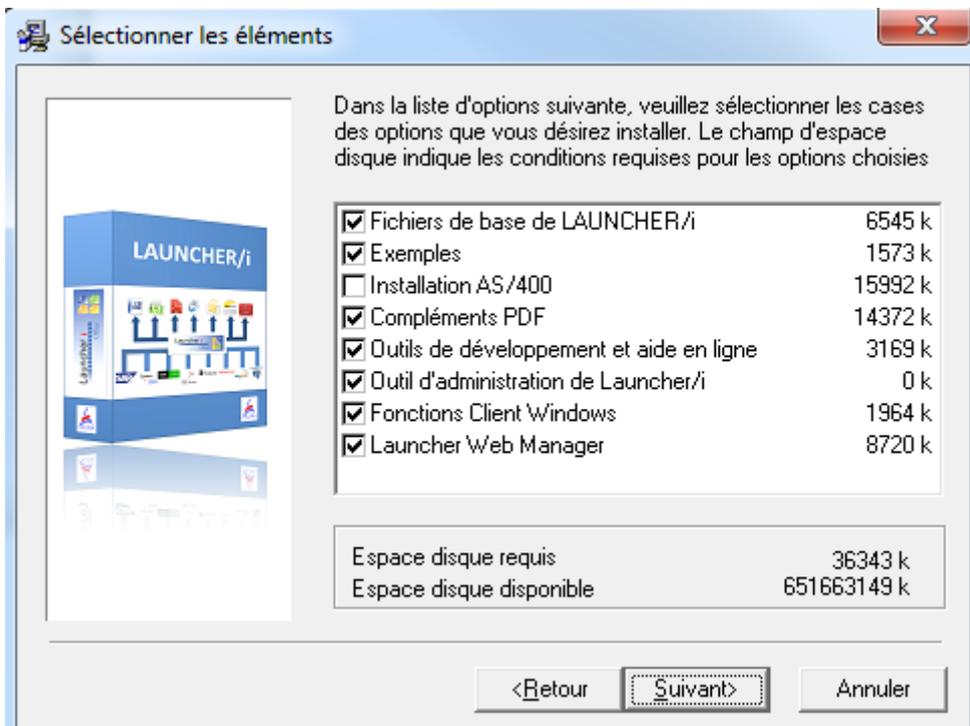
Voir l'exemple de la commande [LNCSNDMAIL - Commande CL](#)

Launcher Web Manager

LAUNCHER Web Manager est une application web permettant de piloter **LAUNCHER Office**. Grâce à cette application vous pouvez offrir à vos utilisateurs un accès simplifié aux fonctionnalités de **LAUNCHER Office**.

Installation

L'installation de l'application web se fait par le biais d'une case à cocher lors de l'installation de LAUNCHER Office.



Une fois installée, et après un redémarrage du PC Windows, l'application est disponible à l'adresse <http://localhost:8080/launcher/>

Pré-requis

Il est nécessaire d'installer l'application Launcher Web Manager sur la même PC Windows que l'application Launcher Office. De plus pour que l'application fonctionne correctement, il est impératif que le processus Launcher Office soit lancé.

L'application Launcher Web Manager utilise le port 8080, ce port doit être libre sur le PC Windows. Afin que l'application soit accessible depuis d'autres machines, il peut-être nécessaire de configurer le pare-feu pour autoriser le port 8080.

Premier accès

Par défaut un seul utilisateur est disponible :

Nom d'utilisateur : admin

Mot de passe : admin

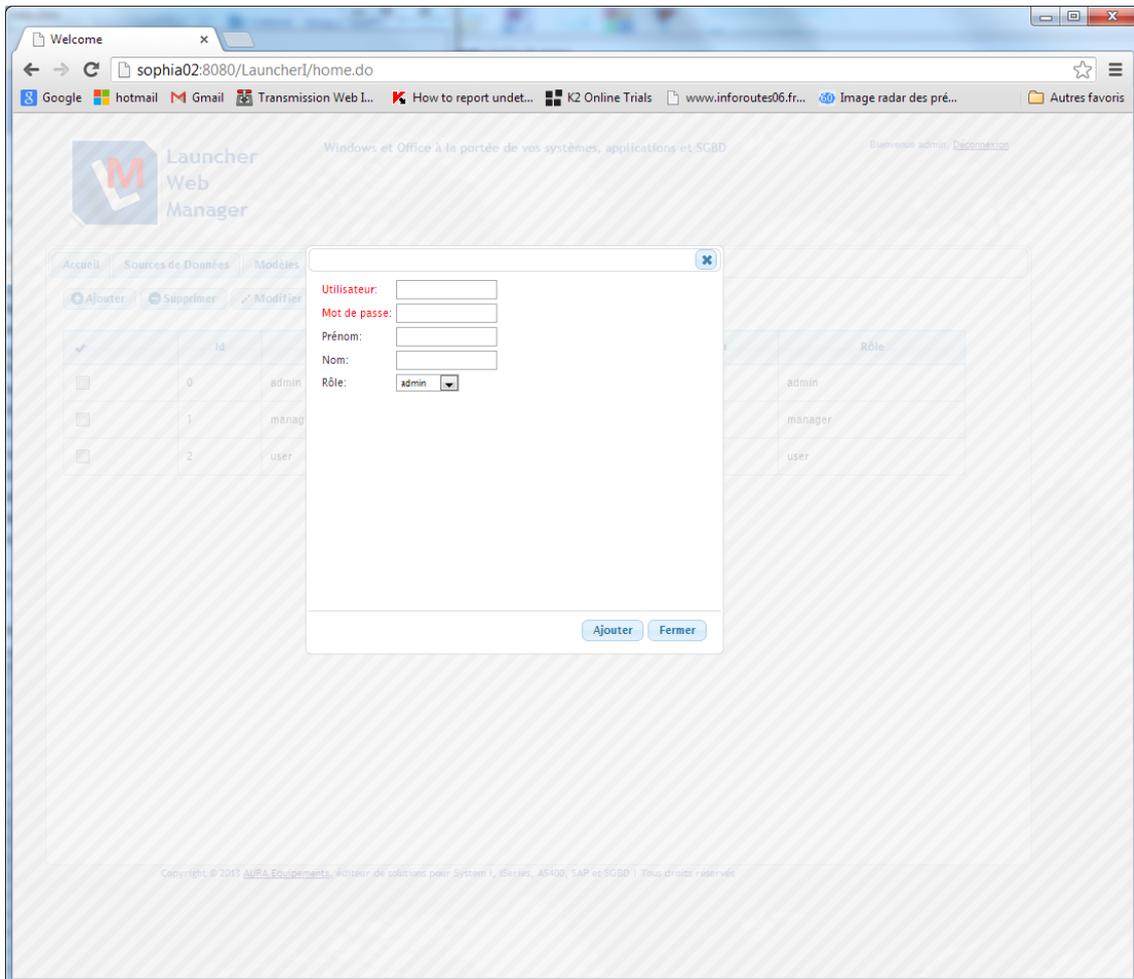
Vous pouvez bien entendu changer ce mot de passe une fois connecté.



Utilisateurs : gestion des utilisateurs

Il est possible pour l'utilisateur admin d'ajouter/supprimer/modifier des utilisateurs. Chaque utilisateur doit obligatoirement avoir un nom, mot de passe et un rôle. Les noms d'utilisateur doivent être unique.

Les champs Nom/Prénom sont optionnels.



Rôles

Il existe trois rôles avec des droits définis comme suit :

'User' : C'est le niveau d'autorisation le plus bas. Les utilisateurs avec ce rôle peuvent uniquement générer de nouveaux documents et les télécharger.

'Manager' : Le rôle 'manager' reprend les droits du rôle 'User', et permet en plus de définir des sources de données et des modèles qui seront utilisés pour générer les documents. Il permet l'accès à ses propres documents et modèles.

Admin : Le rôle 'admin' reprend l'ensemble des droits du rôle 'manager' auquel s'ajoute le droit de gestion des utilisateurs. De plus, l'utilisateur disposant du rôle 'admin' peut visualiser et agir sur tout les models et les documents du système.

Sources de données : gestion des sources de données

Les sources de données représentent les moyen d'obtenir les données qui seront ensuite injectées dans LAUNCHER Office pour la génération de document. Suivant votre version de LAUNCHER Office, vous pouvez créer des sources de données à partir de systemes SAP, base de données Oracle, MySQL, PostgreSQL, MS SQL Server, Sysbase, System i DB2 ou d'un simple fichier CSV.

Source de données SAP

Pour la création d'une source de données SAP, les champs suivant sont obligatoires :

Nom : le nom que vous souhaitez donner à la source de données dans Launcher Web Manager

Adresse IP : adresse ip du serveur SAP au format IPV4

Utilisateur : nom de l'utilisateur SAP

Instance SAP : numéro de l'instance SAP

Client SAP : numéro du client SAP

Server Gateway SAP : adresse ip du serveur gateway SAP, peut être identique à celle du serveur SAP

Service Gateway SAP : numéro de service de la gateway SAP

Il est en possible de créer deux type de sources de données SAP, les sources BAPI ou Table.

Bapi :

Les sources de données BAPI sont des sources dont les données proviennent de l'exécution d'un BAPI SAP. Lors de la définition d'une telle source de données vous devez préciser le nom du BAPI à exécuter ainsi que le nom de la table résultat. Les éventuels paramètres du BAPI seront à remplir lors de la génération du document.

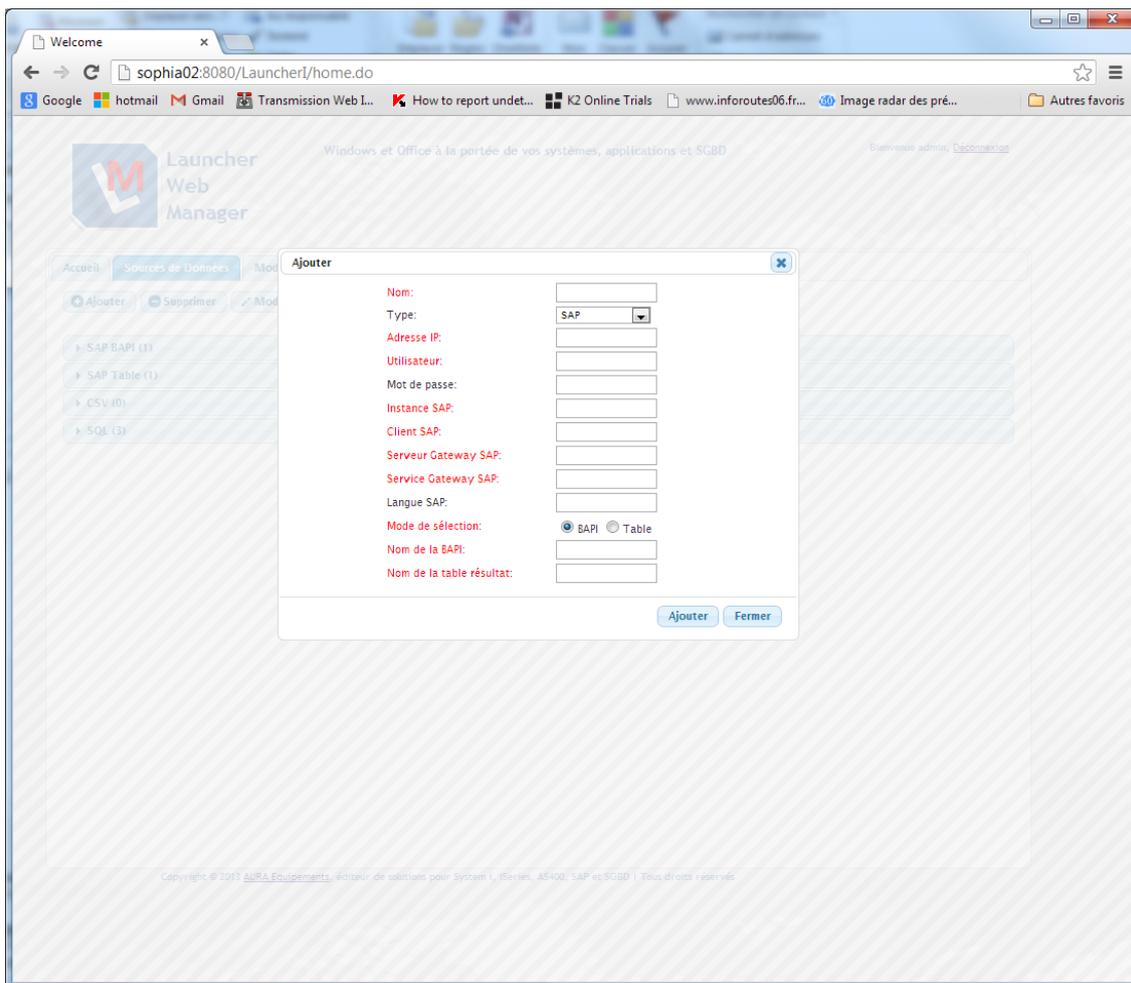
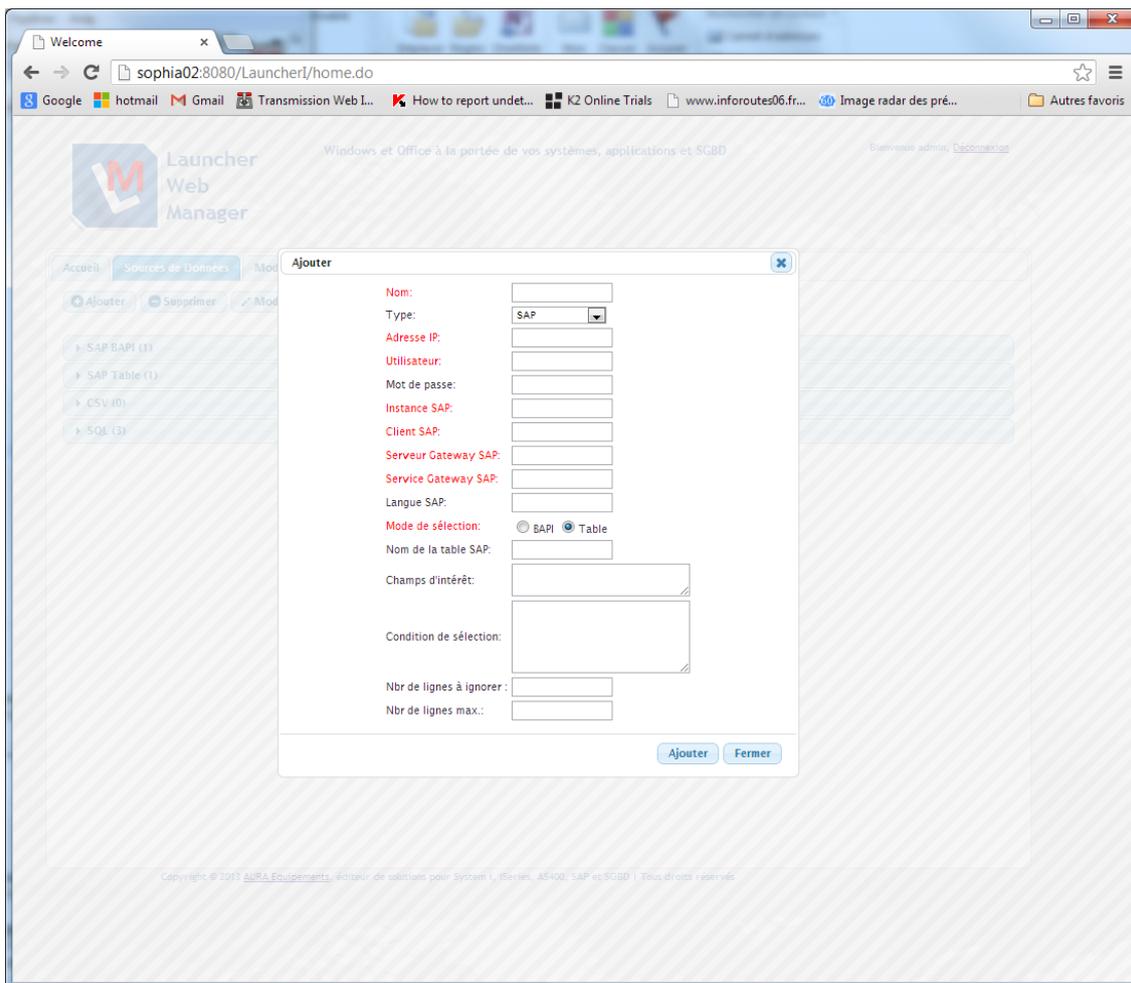


Table :

Dans le cas d'une source de données Table SAP, les données proviennent directement d'un table SAP. Vous devez alors préciser un certain nombre de paramètres supplémentaires : le nom de la table, les champs que vous souhaitez récupérer, la condition de sélection des données, le nombre de ligne à passer et le nombre de ligne maximum.



Source de données SQL

Les sources de données SQL englobent les sources de données Oracle, MySQL, PostgreSQL, MS SQL Server, Sysbase et System i DB2. Pour ces sources de données vous devez obligatoirement spécifier les champs suivants :

Nom : le nom que vous souhaitez donner à la source de données dans Launcher Web Manager

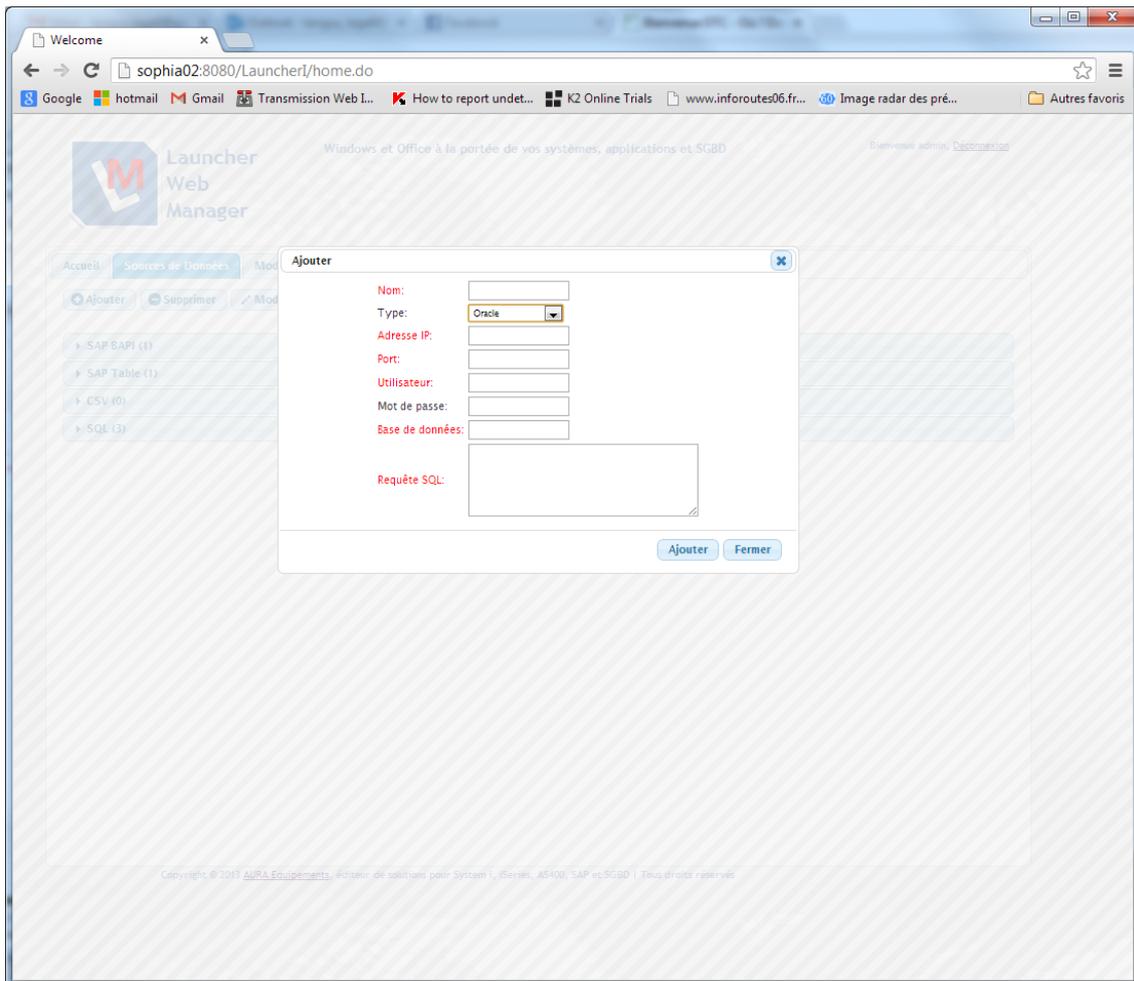
Adresse IP : adresse ip du serveur de base de données au format IPV4

Port : numéro de port correspondant à la base de données

Utilisateur : nom de l'utilisateur de la base de données

Base de données : nom de la base de données ou du schéma.

Requête SQL : la requête SQL servant à récupérer les données.



Source de données CSV

Vous pouvez enfin utiliser une source de données sous le format d'un fichier CSV, dans ce cas les champs suivants sont obligatoires :

Nom : le nom que vous souhaitez donner à la source de données dans Launcher Web Manager

Chemin : chemin complet vers le fichier CSV contenant les données.

Modèles : gestion des modèles de documents

Les modèles de documents sont obligatoirement des fichiers MS Word ou MS Excel, de plus ils doivent être associés à une source de données déjà existante.

Les champs suivants sont obligatoires :

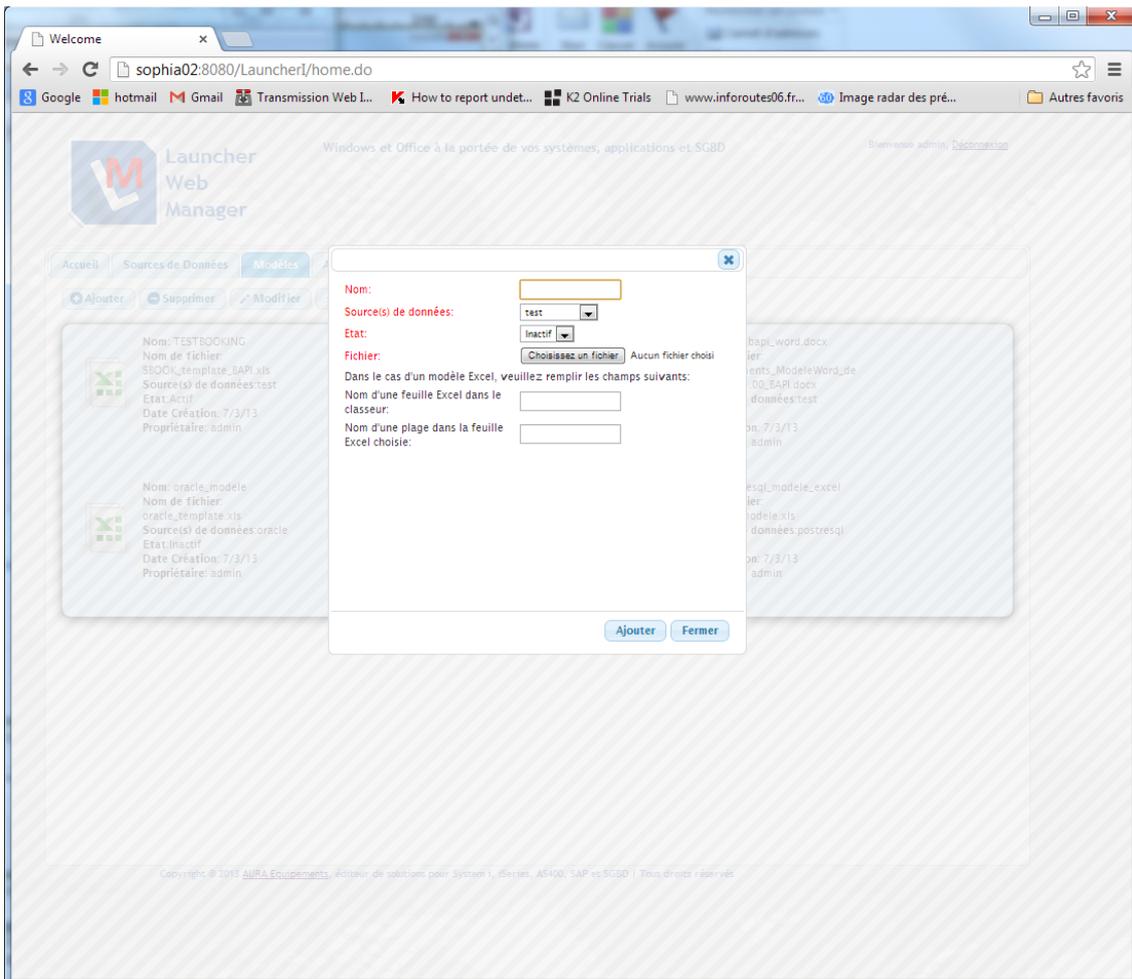
Nom : le nom que vous souhaitez donner au modèle dans Launcher Web Manager

Source de données : le nom de la source de données associée au modèle

Fichier : Le fichier Ms Word ou Ms Excel

Etat : Vous devez spécifier si le modèle est actif ou inactif. Lorsque le modèle est inactif, il n'est pas disponible pour les utilisateurs ayant un rôle 'user' ou pour les autres utilisateurs 'manager' autres que vous-même.

Dans le cas où le modèle est un document MS Excel, vous devez également spécifier le nom de la feuille excel qui va recevoir les données, ainsi que le nom de la plage.



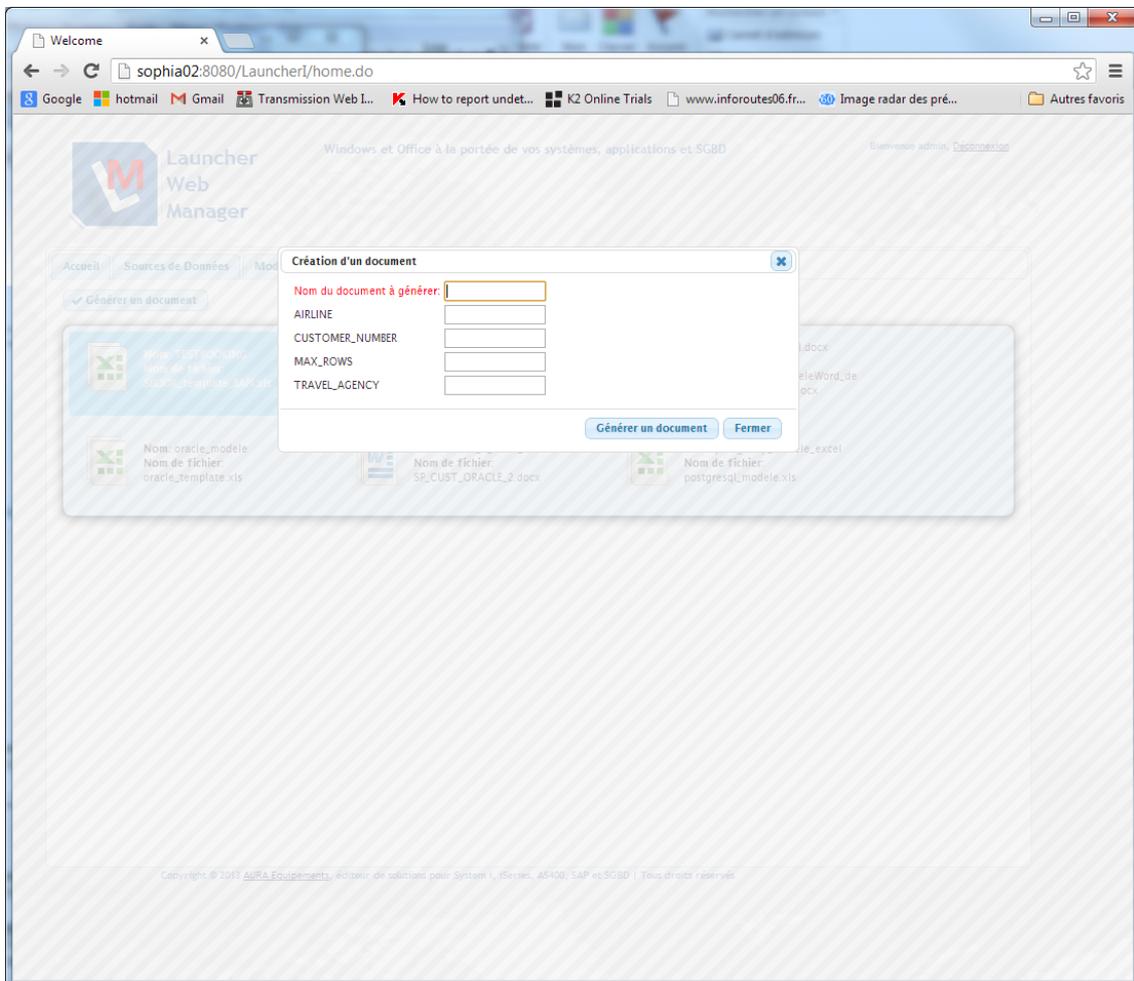
Actions : génération de documents

L'onglet Actions permet la génération de documents à partir de modèles. Cette onglet est accessible à tout les rôles utilisateurs. Les utilisateurs ayant un rôle 'user' peuvent uniquement voir les modèles qui sont actifs. Les utilisateurs ayant un rôle de manager peuvent voir les modèles actifs et les modèles inactifs qui ont été créés par eux-mêmes, ceci afin de pouvoir tester les modèles.

Pour générer un document, il faut sélectionner le modèle puis cliquer sur le bouton générer.

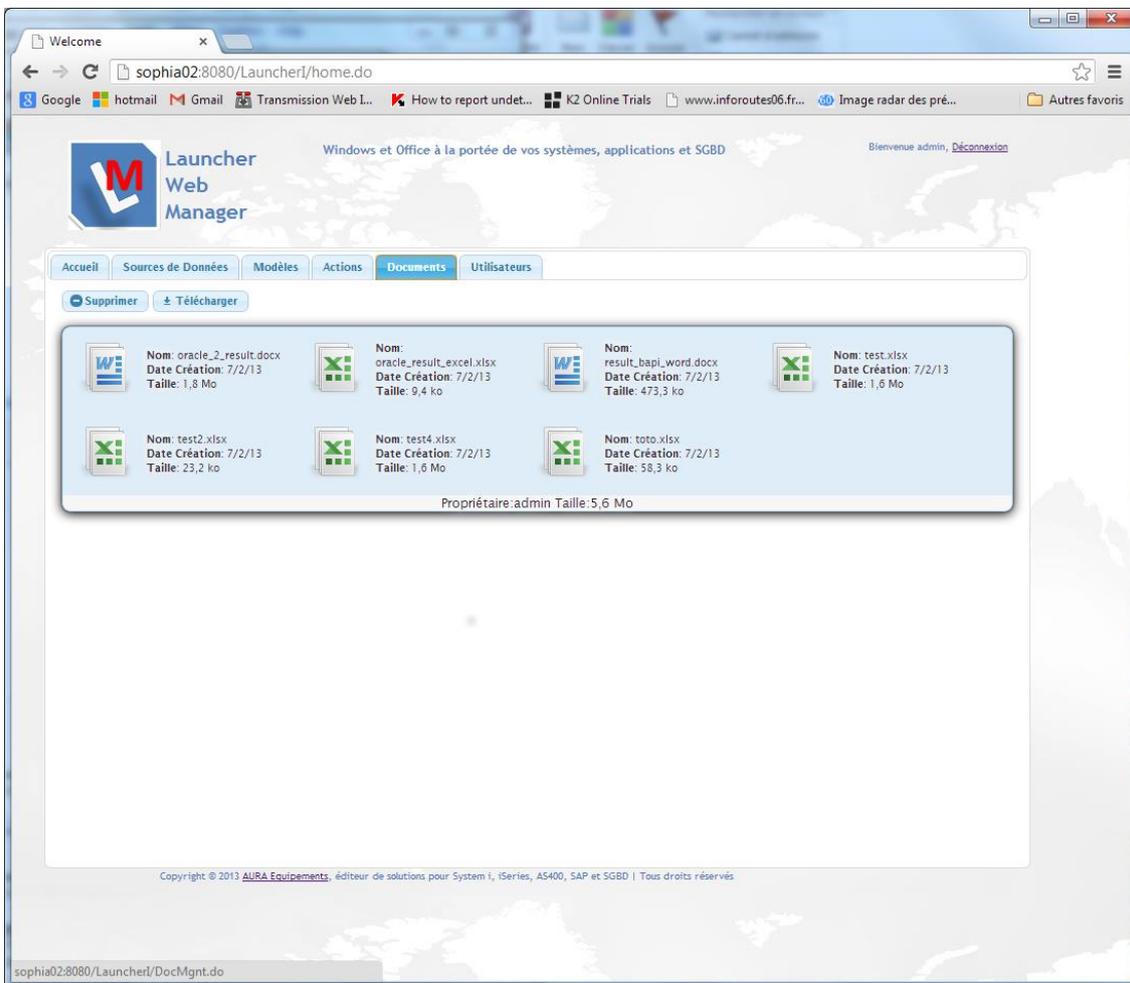
Il faut obligatoirement donner un nom au document final. Les noms de documents doivent être uniques, si le fichier existe déjà il sera remplacé par le nouveau.

Dans le cas d'un modèle basé sur une source de données SAP BAPI, il est proposé lors de la génération du document de remplir les paramètres du BAPI.



Documents : gestion des documents générés

L'onglet 'Documents' permet de gérer les documents générés, de les télécharger ou de les effacer. Les utilisateurs ayant un rôle 'user' ou 'manager' peuvent voir uniquement leurs documents. Les utilisateurs ayant un rôle 'admin' peuvent voir l'ensemble des documents, les télécharger ou les supprimer.



Codes d'erreurs

Erreur de connexion 10061 (274D Hexa) : Connection Refused

Vérifier l'adresse IP, ou le nom de l'AS/400.

Vérifier que le job LAUNCHERD a bien été démarré sur l'AS/400.

La commande pour démarrer le job est : **STRLNCD**

La commande pour arrêter le job est : **ENDLNCD**

LAUNCHER Office utilise par défaut le port numéro **6078**.

LAUNCHERD utilise par défaut le port numéro **5681**.

Erreur de connexion 11001 (2AF9 Hexa) : Host not found

L'AS/400 est désigné par son nom sur le réseau TCP/IP.

Vérifier ce nom. S'il n'a pas été retrouvé par le réseau TCP/IP, ni dans le fichier host du système local, ni sur un serveur DNS.

Vérifier le nom de l'AS/400, les serveurs DNS, ou utiliser une adresse IP de type xxx.xxx.xxx.xxx

Erreur de connexion 10060 (274C Hexa) : Connection Time out

L'adresse TCP/IP contactée n'existe pas sur le réseau.

Vérifier l'adresse TCP/IP de l'AS/400 ou son nom.

Si un nom de machine AS/400 est utilisé, vérifier qu'il est bien référencé sur les serveurs DNS.

Comment nous joindre ?

Pour vos demandes de support technique, merci de créer un ticket via le site internet du support : <http://support.easycom-aura.com>.

N'oubliez pas de vous enregistrer la première fois. Vous recevrez alors votre mot de passe.

Les demandes de support envoyées à l'adresse tech@easycom-aura.com ne sont plus prises en compte.

Pour des informations générales ou commerciales : info@easycom-aura.com.

Copyright

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager AURA Equipements. Le Logiciel décrit dans ce document est régi par un octroi de licence ou accord de confidentialité. Le logiciel ne peut être utilisé, copié ou reproduit sur quelque support que ce soit conformément aux termes de cette licence ou de cet accord de confidentialité. Aucune partie du manuel ne peut être reproduite ou transmise par quelque moyen que ce soit, électronique ou mécanique, y compris par photocopie ou enregistrement, sans la permission expresse et écrite de AURA Equipements.

1986 - 2022 AURA Equipements. Tous droits réservés.

Microsoft, Windows, EXCEL, WORD, Outlook, Exchange sont des marques déposées par Microsoft Corporation.

Lotus Notes est une marque déposée par Lotus Corporation.

IBM, PC/AT, AS/400 sont des marques déposées de International Business Machines Corporation.

Toutes les marques citées sont des marques déposées par leurs auteurs.