

# 4D Internet Commands

---

*Guide de référence*  
*Versions Windows® et Mac™ OS*



---

## **4D Internet Commands**

### **Version 6.7 pour Windows® et Mac™ OS**

*Copyright © 2000 4D SA/4D, Inc.  
Tous droits réservés*

---

Les informations contenues dans ce manuel peuvent faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager 4D SA. La fourniture du logiciel décrit dans ce manuel est régie par un octroi de licence dont les termes sont précisés par ailleurs dans la licence électronique figurant sur le support du Logiciel et de la Documentation y afférente. Le logiciel et sa Documentation ne peuvent être utilisés, copiés ou reproduits sur quelque support que ce soit et de quelque manière que ce soit, que conformément aux termes de cette licence.

Aucune partie de ce manuel ne peut être reproduite ou recopiée de quelque manière que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement, archivage ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur, et ce exclusivement aux conditions contractuelles, sans la permission explicite de 4D SA.

4D, 4D Draw, 4D Write, 4D Insider, 4ème Dimension®, 4D Server, 4D Compiler ainsi que les logos 4e Dimension et 4D sont des marques enregistrées de 4D SA.

Windows, Windows NT et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, Power Macintosh, LaserWriter, ImageWriter, QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2000 est un produit de Altura Software, Inc.

ACROBAT © Copyright 1987-2000, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

Tous les autres noms de produits ou appellations sont des marques déposées ou des noms commerciaux appartenant à leurs propriétaires respectifs.

# Sommaire

## 1. Les commandes Internet de 4D..7

Préface.....	9
Installation et logiciels requis.....	11
Glossaire et terminologie.....	13
Format des paramètres.....	16

## 2. IT Send Mail..... 19

Envoi de courrier, Présentation.....	21
SMTP_SetPrefs.....	23
SMTP_GetPrefs.....	26
SMTP_QuickSend.....	27
SMTP_New.....	29
SMTP_Host.....	30
SMTP_Send.....	31
SMTP_Clear.....	32
SMTP_Date.....	33
SMTP_From.....	36
SMTP_Sender.....	38
SMTP_ReplyTo.....	40
SMTP_To.....	42
SMTP_Cc.....	43
SMTP_Bcc.....	44
SMTP_InReplyTo.....	46
SMTP_References.....	47
SMTP_Comments.....	48
SMTP_Keywords.....	50
SMTP_Encrypted.....	51
SMTP_AddHeader.....	52
SMTP_Subject.....	54
SMTP_Body.....	56
SMTP_Attachment.....	59
SMTP_Charset.....	61

### 3. IT Review Mail.....63

Réception de courrier, Présentation.....	65
POP3_SetPrefs.....	67
POP3_GetPrefs.....	68
POP3_Login.....	69
POP3_VerifyID.....	71
POP3_Reset.....	72
POP3_Delete.....	73
POP3_Logout.....	74
POP3_BoxInfo.....	75
POP3_MsgInfo.....	76
POP3_GetMessage.....	77
POP3_MsgLstInfo.....	78
POP3_MsgLst.....	80
POP3_Download.....	82
POP3_UIDToNum.....	84
POP3_Charset.....	85

### 4. IT Downloaded Mail.....87

Téléchargement de courrier, présentation.....	89
MSG_FindHeader.....	90
MSG_MessageSize.....	91
MSG_GetHeaders.....	92
MSG_GetBody.....	93
MSG_GetMessage.....	94
MSG_HasAttach.....	95
MSG_Extract.....	96
MSG_Delete.....	97

### 5. IT File Transfer.....99

Transfert de fichiers, Présentation.....	101
FTP_Progress.....	102
FTP_Login.....	104
FTP_GetDirList.....	106

FTP_ChangeDir.....	108
FTP_PrintDir.....	109
FTP_GetFileInfo.....	110
FTP_VerifyID.....	111
FTP_MakeDir.....	112
FTP_RemoveDir.....	113
FTP_Rename.....	114
FTP_Delete.....	115
FTP_MacBinary.....	116
FTP_Send.....	118
FTP_Append.....	120
FTP_GetType.....	121
FTP_SetType.....	122
FTP_System.....	123
FTP_Receive.....	124
FTP_Logout.....	126
FTP_SetPassive.....	127
FTP_GetPassive.....	128

## 6. IT TCP/IP..... 129

Routines de bas niveau, Présentation.....	131
TCP_Open.....	132
TCP_Listen.....	133
TCP_Send.....	135
TCP_Receive.....	136
TCP_SendBLOB.....	137
TCP_ReceiveBLOB.....	138
TCP_State.....	139
TCP_Close.....	140

## 7. IT Internet..... 141

Commandes spéciales Internet, Présentation.....	143
NET_Finger.....	144
NET_Ping.....	146
NET_Time.....	147
NET_NameToAddr.....	149

NET_AddrToName.....	150
NET_Resolve.....	151

## 8. IT Utilities..... 153

Utilitaires, Présentation.....	155
IT_MacTCPInit.....	156
IT_Platform.....	157
IT_Version.....	158
IT_TCPVersion.....	159
IT_MacTCPVer.....	160
IT_MyTCPAddr.....	162
IT_SetTimeOut.....	163
IT_GetTimeOut.....	164
IT_ErrorText.....	165
IT_Encode.....	166
IT_Decode.....	168
IT_GetProxy.....	170
IT_SetProxy.....	171
IT_GetPort.....	173
IT_SetPort.....	174

## 9. Annexes..... 175

Annexe A, Conseils de programmation.....	177
Annexe B, Numéros des ports TCP.....	179
Annexe C, Codes d'erreurs de 4D Internet Commands.....	181
Annexe D, Informations supplémentaires.....	190

## Index des commandes..... 191

## **Les commandes Internet de 4D**





Les commandes Internet de 4D ("4D Internet Commands") offrent aux utilisateurs de 4e Dimension un ensemble puissant d'outils de communication permettant de travailler sur tout type de réseau, local, national ou mondial. Ces dernières années, le nombre d'utilisateurs et d'entreprises accédant à Internet s'est considérablement accru. Avec la croissance du nombre d'internautes, le besoin d'être présent "sur le net" est plus fortement ressenti chaque jour par les professionnels.

Les commandes Internet de 4D fournissent aux développeurs de bases de données 4D un accès à de nombreux éléments-clés d'Internet. Les commandes SMTP contiennent des outils permettant d'automatiser l'envoi de messages électroniques depuis une base de données vers une liste illimitée de destinataires. De même, les commandes POP3 permettent de récupérer du courrier d'un nombre illimité de boîtes aux lettres électroniques pour le stocker dans une base de données, le réacheminer, lui apporter une réponse automatique ou effectuer une recherche à distance. Les commandes FTP permettent de transférer des fichiers depuis/vers des systèmes distants ou encore de lister les documents et répertoires présents sur un volume FTP. Enfin, les commandes TCP fournissent aux développeurs des outils de bas niveau pour leur permettre d'exécuter de multiples tâches liées à Internet.

Le protocole SMTP (Simple Mail Transfer Protocol) est le principal protocole de transfert de courrier utilisé sur Internet. Les commandes Internet de 4D permettent de créer et d'envoyer rapidement du courrier via un serveur SMTP. La création et l'envoi de courrier peuvent s'effectuer à l'aide d'une seule commande. Si vos besoins en matière de messagerie sont plus complexes, les commandes Internet de 4D fournissent des outils permettant de contrôler entièrement les en-têtes, le corps et les pièces jointes des messages. Comme le courrier Internet peut être adressé à des réseaux "privés" tels que CompuServe, America Online, eWorld, etc., vous êtes donc en mesure d'atteindre virtuellement quiconque possédant un compte e-mail. Le jeu de commandes SMTP permet également, par exemple :

- l'envoi automatisé de statistiques et d'états depuis vos bases de données ;
- la création d'une base de données de réacheminement automatique de courrier ;
- la gestion d'une liste de diffusion de groupe (*mailing list*) ;
- la synchronisation de bases de données distantes.

Outre les commandes SMTP, les commandes Internet de 4D contiennent également des outils permettant la connexion à des serveurs de courrier électronique POP3 (Post Office Protocol, Version 3) pour récupérer des messages électroniques et des documents joints encodés. Les commandes SMTP et POP3 sont conformes aux normes MIME s'appliquant aux pièces jointes multiples, ce qui en facilite le téléchargement et la sauvegarde.

Des commandes permettent également aux utilisateurs d'encoder les documents joints sous divers formats tels que Binhex, Base64, AppleSingle, AppleDouble...

Les commandes FTP (File Transfer Protocol) sont très simples d'emploi pour envoyer des fichiers texte ou binaires à un serveur FTP et en recevoir. Les commandes FTP permettent d'obtenir la liste des répertoires et des fichiers, facilitant ainsi le développement d'interfaces de navigation vers des volumes distants. On peut facilement intégrer des commandes FTP dans des applications de recherche de documents sans devoir "monter" les volumes distants sur le poste client.

Le protocole TCP/IP (Transmission Control Protocol/Internet Protocol) est le principal protocole utilisé pour l'envoi et la réception de données sur Internet. Plusieurs commandes Internet de 4D permettent d'envoyer et de recevoir des paquets TCP. Les commandes TCP fournissent aux développeurs les outils nécessaires à la construction et au contrôle de leurs communications sur Internet.

Par exemple, vous pourrez ainsi :

- construire votre propre interface telnet ;
- exécuter des instructions sur des machines distantes ;
- récupérer des documents sur le World Wide Web ;
- effectuer des recherches dans de multiples bases de données en ligne ;
- gérer la synchronisation de bases de données avec des serveurs distants ;
- suivre des colis confiés à Federal Express ou UPS.

#### Installation

---

Les commandes Internet de 4D sont installées dans 4e Dimension par l'intermédiaire d'un plug-in, appelé "4D Internet Commands".

4D Internet Commands est disponible dès que vous installez 4e Dimension ou 4D Server (à compter de la version 6.5). Il est copié à deux endroits différents :

- dans le dossier Système actif (Windows\4D\WIN4DX sous Windows et Dossier Système:Préférences:4D:MAC4DX sous Mac OS), ce qui le rend immédiatement disponible dans toutes vos applications 4D.
- dans le dossier d'installation standard.

Vous pouvez donc choisir d'installer le plug-in 4D Internet Commands :

- dans certaines bases de données uniquement — dans ce cas, il vous suffit de le placer dans un dossier Mac4DX ou Win4DX au même niveau que la structure de ces bases de données,
- dans toutes vos bases de données — dans ce cas, il vous suffit de le laisser dans le dossier Système actif.

#### Notes importantes sur l'installation :

- Si 4D Internet Commands est installé aux deux endroits, seule la version située dans le dossier WIN4DX/MAC4DX au même niveau que la structure de la base de données est chargée.
- Compilation : lors de la compilation d'une base de données qui utilise 4D Internet Commands installé dans le dossier Système, il est nécessaire d'indiquer explicitement à 4D Compiler le chemin d'accès du plug-in 4D Internet Commands.
- Lors de la copie ou du déplacement d'une base de données ou d'un exécutable vers une autre machine, assurez-vous que le dossier WIN4DX/MAC4DX dans lequel se trouve le plug-in est copié en même temps.

Pour plus d'informations, veuillez consulter le *Guide d'installation* de 4D Product Line.

#### Configuration requise

---

La liste suivante présente les logiciels requis pour utiliser les commandes Internet de 4D dans une base de données 4e Dimension, en fonction du jeu de commandes que vous souhaitez utiliser.

Pratiquement toutes les commandes Internet de 4D communiquent via le protocole TCP/IP, vous devez donc disposer au minimum de ce protocole.

### **Matériel**

- Macintosh à processeur 680x0 ou PowerPC
- PC compatible Intel sous Windows 95/98 ou Windows NT 2000

### **4e Dimension (Mac & Windows)**

- 4D MacOS version 6.0 ou ultérieure
- 4D Windows version 6.0 ou ultérieure

### **MacTCP, Open Transport, Winsock**

Comme la plupart des commandes Internet de 4D utilisent le protocole TCP/IP pour leurs communications, un ordinateur devant exécuter ces commandes doit disposer d'un gestionnaire TCP/IP installé et correctement configuré, avec une adresse IP unique. La plupart des systèmes d'exploitation ont un gestionnaire TCP/IP pré-installé, par exemple MacTCP ou Open Transport sous MacOS et Winsock sous Windows.

Pour plus d'informations sur la configuration du protocole TCP/IP, veuillez faire appel à votre administrateur réseau ou consulter le manuel *Composants réseau pour 4D Server*.

**Note :** A compter de la version 6.5, les composants réseau 4D sous MacOS fonctionnent uniquement avec l'architecture "Open Transport" (68K ou PPC). Toutefois, les commandes Internet de 4D n'utilisent pas le composant réseau 4D TCP/IP pour communiquer avec le protocole TCP/IP, elles peuvent donc utiliser le gestionnaire MacTCP.

En revanche, si vous voulez utiliser la version 6.5 (et ultérieure) de 4D Client ou les fonctions Web de 4e Dimension 6.5 (et ultérieure), vous devez disposer du gestionnaire TCP/IP "Open Transport" sous MacOS.

### **Accès au réseau**

Pour pouvoir utiliser les commandes Internet de 4D, vous devez avoir accès à un réseau exploitant le protocole TCP/IP.

### **Serveur de noms de domaine**

Pour de nombreuses commandes Internet de 4D, il est nécessaire d'avoir accès à un serveur de noms de domaines (*Domain Name Server*, ou DNS).

### **Serveur de courrier électronique SMTP**

Pour envoyer du courrier au moyen des commandes SMTP, l'expéditeur doit avoir accès à un serveur de courrier électronique SMTP.

### **Serveur de courrier électronique POP3**

Pour utiliser les commandes POP3, vous devez avoir un compte sur un serveur de courrier électronique POP3.

Cette partie définit succinctement la plupart des concepts et termes spécifiques utilisés dans ce manuel. La section suivante, "Format des paramètres", fournit des informations supplémentaires sur les paramètres des commandes Internet de 4D.

**NIC** : "Network Information Center" (Centre d'informations sur le réseau). Internet est, de manière générale, une entité non réglementée. Il n'existe pas d'autorité centralisant ou contrôlant son utilisation et son développement. Toutefois, il est nécessaire qu'un organisme unique régule les attributions de noms de domaine et d'adresses IP. Le NIC remplit ce rôle.

**RFC** : "Request for Comments". La plupart des commandes Internet de 4D s'appuient sur des normes définies pour traiter les communications sur Internet. Les méthodologies, descriptions et protocoles standard utilisés sur Internet sont définis dans des documents intitulés RFC. L'Annexe D, Informations supplémentaires référence les sites Web comportant des liens vers la plupart des documents RFC. Bien que les commandes Internet de 4D simplifient la programmation des accès à Internet, il peut parfois vous être utile de consulter certains de ces documents, en particulier si vous souhaitez utiliser les routines de communication TCP de bas niveau.

**Adresses TCP/IP, noms de serveur et noms de domaine** : L'adresse IP est la référence d'une machine **spécifique** se trouvant quelque part dans le monde. Une adresse IP se présente sous la forme d'une chaîne de caractères contenant quatre valeurs numériques séparées par des points (par exemple, "207.94.15.3"). Chaque partie numérique de l'adresse peut contenir une valeur comprise entre zéro et 255. En appliquant certaines fonctions mathématiques à une adresse IP, sa valeur peut être exprimée sous forme d'entier long (appelée **ip\_EntierLong** dans ce manuel).

Afin que les ordinateurs d'un site (par exemple une entreprise, une université, etc.) puissent se connecter à Internet, des garanties doivent être prises pour s'assurer que leurs adresses IP ne rentrent pas en conflit avec d'autres machines sur le réseau. Les sociétés (et parfois les particuliers) peuvent donc inscrire leur site auprès du **NIC** afin d'obtenir un **nom de domaine**. L'utilisation de **noms de domaines** (par exemple "www.4d.com" ou "ftp.4d.com") facilite l'identification et la lecture des adresses Internet. Les noms de domaines sont traduits par le DNS (cf. ci-dessous) en adresses numériques (numéros IP) utilisées par le réseau.

**Nom de domaine** = "4d.com"

<b>Nom d'hôte (Nom de serveur)</b>	=	<b>Adresse IP</b>	=	<b>ip_EntierLong</b>
"www.4d.com"	=	"207.94.15.3"	=	-815919357

La correspondance entre un **Nom d'hôte** et son adresse IP est stockée dans une base de données appelée DNS (*Domain Name System* ou Système de noms de domaine). Les DNS communiquent entre eux pour échanger toute donnée nouvelle ou modifiée dans les listes de noms de domaines du monde entier. Le panneau de contrôle TCP/IP permet de "faire pointer" votre ordinateur sur un DNS, qui se chargera alors de traduire les références des noms de domaine que vous utilisez.

Il est important de noter que tous les serveurs de noms de domaines disposent d'une adresse IP. Cependant, toutes les adresses IP n'ont pas un serveur de noms de domaine correspondant. De même, une adresse électronique telle que "jsmith@4d.com" ne désigne pas l'ordinateur spécifique ou l'adresse IP de cette personne. L'adresse électronique dirige la distribution du courrier vers la machine dont l'adresse IP est obtenue en convertissant le domaine "4d.com". Le courrier est distribué au serveur POP3 fonctionnant sur cette machine, qui le met alors en attente pour l'utilisateur nommé "jsmith".

**Nom de domaine :** Le nom de domaine est une structure d'adressage utilisée pour l'identification et la localisation des ordinateurs sur Internet. Les noms de domaines facilitent la mémorisation des adresses Internet, qui sont traduites par le système de noms de domaines (DNS) en adresses numériques (numéros Internet Protocol [IP]) utilisées par le réseau. Un nom de domaine est structuré et fournit souvent des informations sur le type d'entité qu'il représente. Les noms de domaines situés au même niveau hiérarchique doivent être uniques. Par exemple, il ne peut y avoir qu'un seul com au niveau supérieur de la hiérarchie et un seul 4d.com au niveau suivant de la hiérarchie. Si le nom de votre entreprise est "NomEntreprise", vous pourriez enregistrer le nom de domaine "NomEntreprise.com" et votre adresse de courrier électronique pourrait être "NomUtilisateur@NomEntreprise.com". Vos clients pourraient accéder à votre site Web en se connectant à "www.NomEntreprise.com" avec leur navigateur Web.

**Système de noms de domaine (DNS) :** Il s'agit d'une base de données répartie contenant des informations utilisées pour traduire des noms de domaines, qui sont faciles à mémoriser et à utiliser, en numéros Internet Protocol (IP), nécessaires pour localiser les ordinateurs sur Internet. Les utilisateurs du monde entier conservent et mettent jour la partie de la base de données qui les concerne, et la totalité de cette base est disponible pour tous les ordinateurs et utilisateurs d'Internet. Le DNS comprend les ordinateurs, les fichiers de données, les logiciels et les personnes travaillant ensemble.

**Encodage :** L'encodage est utilisé pour convertir un fichier dans un format interprétable par tout type de système d'exploitation (ASCII standard). Le type d'encodage le plus répandu est l'encodage binaire-héxadécimal (Binhex). C'est l'option d'encodage par défaut de la plupart des documents que vous joignez à vos messages. Un fichier encodé est plus gros que l'original.

Sous MacOS, l'encodage permet de convertir la partie "données" (*data fork*) et la partie "ressources" (*resource fork*) d'un fichier en un document de type texte qui peut être facilement envoyé comme document joint. Les commandes Internet de 4D acceptent les modes d'encodage les plus courants dont Binhex, Base64, AppleSingle, AppleDouble, UUEncode et MacBinary.

**Encryptage** : L'encryptage est utilisé pour brouiller intentionnellement le contenu des messages. Les messages sont codés au moyen d'un programme externe de cryptage, tel que PGP, dans le seul but d'accroître leur confidentialité. Le texte encrypté doit alors être décrypté avant d'être lu. Les commandes Internet de 4D **ne fournissent pas** de moyen d'encrypter des messages.

**Compression** : La compression est utilisée pour réduire l'espace disque occupé par un fichier. Pour compresser un fichier, vous pouvez utiliser une application telle que Stuffit Deluxe™ Compact Pro™ ou WinZip™. Le fichier doit ensuite être décompressé par l'application qui l'a créé pour retrouver son format original. Les applications de compression ajoutent généralement un suffixe au nom original du fichier. Voici quelques suffixes courants et leurs applications respectives :

*Nomfichier*.SIT - application Stuffit

*Nomfichier*.CPT - application Compact Pro

*Nomfichier*.DD - application Disk Doubler

*Nomfichier*.ZIP - application Winzip

*Nomfichier*.SEA - Self Extracting Archive. Ces fichiers sont des applications Macintosh auto-extractibles, ils se décompressent seuls lorsque l'utilisateur double-clique dessus car le code de décompression est inclus. En raison de l'ajout de ce code, les archives auto-extractibles sont généralement plus volumineuses que les autres. Cependant, puisque l'utilisateur n'a pas besoin de l'application de compression, cette option peut s'avérer avantageuse.

Il est important de noter qu'une fois compressé, un fichier doit être encodé afin de pouvoir être transmis correctement de machine en machine jusqu'à sa destination finale.

Cette section indique la signification et le formatage des paramètres clés utilisés dans ce manuel.

Paramètre	Type	Description
nomServeur	Alpha	→ Nom du serveur (Ex. : "www.nomdesociete.com") Adresse IP (Ex. : "204.118.90.2")
ip_EntierLong	Entier long	→ Référence d'une adresse IP sous forme d'entier long
adresseEmail	Texte	→ Ex. : "jsmith@4d.com"
listeAdresses	Texte	→ Ex. : "jsmith@4d.com,jdupont@4d.fr" ou "john@4d.com"+Caractere(13)+"jean@4d.fr"
cheminLocal	Texte	→ • Document Mac : "Disque dur:BDD:Ventes:Rapport" Win : "C:\Dossier\BDD\Ventes\Rapport.txt" • Répertoire Mac : "Disque dur:En cours:" (Notez le ":" final) Win : "C:\EnCours\" (Notez le "\" final)
cheminServeur	Texte	→ • Document "/usr/jsmith/rapports/rapportventes.txt" • Répertoire "/usr/jsmith/rapports/"(Notez le "/" final)
tcp_ID	Entier long	→ Référence d'une session TCP ouverte
smtp_ID	Entier long	→ Référence d'un nouveau message électronique
pop3_ID	Entier long	→ Référence d'une session POP3 ouverte
ftp_ID	Entier long	→ Référence d'une session FTP ouverte
Résultat	Entier	← Code d'erreur

### nomServeur

Le paramètre nomServeur est le nom ou l'adresse IP du serveur hôte (*HostName*), par exemple "dns.4d.com" ou "204.118.90.2". Les noms de serveurs sont convertis au moyen du système de noms de domaines. Les noms de domaines "par défaut" (*primaires*) et "secondaires" sont généralement indiqués dans le tableau de bord du gestionnaire TCP/IP installé. Les commandes Internet de 4D nécessitant un nom de serveur comme paramètre acceptent indifféremment son nom ("www.4d.com") ou son adresse IP ("204.118.90.2"). Le format "nom" est généralement préférable car il met l'application à l'abri d'effets indésirables liés aux modifications matérielles dans les sites distants.



### ip\_EntierLong

Des formules mathématiques peuvent être appliquées aux adresses IP pour les convertir en entiers longs uniques. Les commandes NET\_NameToAddr et NET\_AddrToName (thème 'IT Internet') automatisent cette conversion. Cet entier long est désigné par ip\_EntierLong dans ce manuel. Cette valeur n'est utilisée que dans des circonstances particulières par des développeurs établissant des communication TCP directes.

Certains développeurs préféreront également stocker des entiers long plutôt que des noms de domaines afin d'économiser de l'espace disque. Toutefois, pour des raisons de compatibilité avec IPV6, cette astuce est déconseillée. Il est préférable de travailler avec l'adresse IP ou le nom de domaine.

### adresseEmail

Le paramètre adresseEmail est une spécification d'adresse électronique complète au format "nom\_utilisateur@nom\_domaine". Dans ce manuel, adresseEmail fait référence à une **adresse électronique unique**. Lorsqu'un paramètre d'une routine accepte une liste de plusieurs adresses, le paramètre listeAdresses est explicitement indiqué. adresseEmail ne peut accepter qu'une seule adresse électronique. Il doit comporter à la fois le nom de l'utilisateur et le nom de domaine :

"Felix Unger" <felix@pristine.com>  
oscar@slobs.com (Oscar Madison)

### listeAdresses

Le paramètre listeAdresses contient **une ou plusieurs adresses électroniques** au format décrit dans le paramètre adresseEmail, les adresses étant séparées par des virgules ou des retours chariot. La délimitation par retour chariot est utile pour fournir aux utilisateurs une zone de texte permettant de saisir ou de coller plusieurs adresses. Les trois exemples suivants génèrent une valeur \$listeAdresses valide:

```
$ListeAdresses:="jsmith@4d.com"
```

```
$ListeAdresses:="jsmith@4d.com,scott@4d.com,marcel@4d.fr"
```

**Boucle (\$i;1;Taille tableau(aAdresses))**

```
$ListeAdresses:=$ListeAdresses+aAdresses{$i}+Caractere(13)
```

**Fin de boucle**

### cheminLocal

Le paramètre cheminLocal indique l'emplacement d'un fichier ou d'un répertoire sur l'ordinateur de l'utilisateur (Mac ou Windows). Sur un Macintosh, les éléments à l'intérieur des dossiers sont séparés par des caractères "deux-points" (:). Par exemple, le fichier "Mon rapport" dans le dossier "Rapports" sur le disque dur "Mon disque dur" aura comme chemin d'accès "Mon disque dur:Rapports:Mon Rapport". Une spécification de répertoire sur un Macintosh doit se terminer par un caractère "deux-points".

Par exemple, si vous voulez placer un nouveau rapport dans le dossier indiqué précédemment, vous devez passer la chaîne "Mon disque dur:Rapports:". La décision de faire référence à un nom de fichier ou de répertoire est liée au contexte de la commande. Sous Windows, le principe utilisé est identique, à l'exception du fait qu'une barre oblique inversée "\" est utilisée à la place des caractères "deux-points".

### **cheminServeur**

Le cheminServeur est l'emplacement d'un fichier ou d'un répertoire sur un ordinateur fonctionnant sous le système d'exploitation Unix. Dans l'environnement Unix, les répertoires sont séparés par des barres obliques ("/"). Par exemple, le fichier "rapport.txt" dans le répertoire "rapports" du répertoire "4D" sera désigné par "/4D/rapports/rapport.txt". Le chemin d'accès d'un répertoire doit se terminer par un caractère "/". Notez qu'un chemin d'accès complet commence par une "/" qui représente la racine du volume.

### **smtp\_ID, pop3\_ID, ftp\_ID, tcp\_ID**

La plupart des commandes Internet de 4D, quel que soit leur thème, utilisent un numéro d'ID permettant de référencer de façon unique leur session de travail. Toutes les commandes qui doivent s'exécuter dans le cadre d'une session y font référence par l'intermédiaire de l'ID de la session.

Les numéros d'"ID" sont propres à chaque classe de commande (SMTP, POP3, FTP, TCP). Par exemple, un numéro d'ID POP3 ne peut pas être passé en tant que paramètre à une commande TCP, ou encore une variable ftp\_ID identifiant une session FTP ne peut pas être passée aux routines TCP.

<b>Référence de session</b>	<b>Ouverte par</b>	<b>Fermée par</b>
tcp_ID	TCP_Open ou TCP_Listen	TCP_Close
smtp_ID	SMTP_New	SMTP_Clear
pop3_ID	POP3_Login	POP3_Logout ou POP3_VerifyID
ftp_ID	FTP_Login	FTP_Logout ou FTP_VerifyID

### **Résultat**

Toutes les commandes Internet de 4D (à l'exception de IT\_ErrorText et IT\_Version) retournent un entier comme résultat de la fonction. Cet entier contient un numéro d'erreur que la commande doit retourner à la base de données 4D.

Si une commande aboutit, un zéro est renvoyé. Sinon, un code d'erreur est renvoyé.

L'Annexe C, Codes d'erreurs de 4D Internet Commands fournit la liste des codes d'erreurs des commandes Internet de 4D.

# 2

---

## IT Send Mail



SMTP (Simple Mail Transport Protocol) est le protocole standard du courrier électronique sur Internet. Avec les commandes Internet de 4D, vous pouvez créer des messages électroniques simples au moyen d'une seule commande, ou des messages complexes grâce à une série de commandes. Les commandes SMTP vous permettent de contrôler toutes les parties d'un message électronique, dont les en-têtes de Réponse à (*Reply-To*) et d'Expéditeur (*Sender*), les pièces jointes, les commentaires et les références.

L'intégration des commandes Internet dans 4e Dimension vous permet de développer des bases de données très puissantes pouvant envoyer des messages et des documents sur Internet. A l'aide des commandes SMTP, vous pourrez notamment :

- automatiser l'envoi d'états ou de documents créés dans 4e Dimension,
- construire des applications 4D informant automatiquement les développeurs lors d'événements spécifiques, par exemple en cas d'incident (en utilisant APPELER SUR ERREUR),
- envoyer des mailings automatiquement à des personnes dans le monde entier.

Ces commandes, associées aux commandes POP3 (récupération de fichiers et de documents joints), FTP et TCP, fournissent au développeur les outils lui permettant d'augmenter de façon considérable les capacités de communications de ses bases de données 4D.

### Deux méthodes de création d'un message électronique

Les commandes SMTP permettent d'envoyer du courrier électronique de deux manières distinctes, appelées précédemment méthode "simple" et méthode "complexe". La méthode "simple" implique une seule commande, SMTP\_QuickSend, qui accepte tous les paramètres nécessaires à l'adressage et à l'envoi d'un message.

La majorité des messages envoyés dans le monde sont plutôt simples dans leur construction : quelqu'un "ici" souhaite envoyer un "message" quelconque à quelqu'un se trouvant "là-bas" à propos d'un "objet" spécifique. S'il s'agissait d'un courrier classique, vous devriez écrire le texte, fermer l'enveloppe, rédiger l'adresse puis porter la lettre au bureau de poste pour qu'elle soit expédiée. Avec SMTP\_QuickSend, vous pouvez préciser l'Emetteur (*From*), le Destinataire (*To*), l'Objet (*Subject*) et le Corps du message (*Message Body*) au moyen d'une seule commande.

Cependant, certains messages peuvent nécessiter des paramétrages plus complexes. Par exemple, supposons que la lettre mentionnée précédemment doive être envoyée en plusieurs exemplaires à d'autres destinataires ou qu'un document comme votre rapport annuel doive lui être joint. Dans ce cas, vous devriez photocopier la lettre, imprimer des rapports et préparer une enveloppe pour chaque destinataire. Les commandes SMTP de 4D simplifient la distribution en vous permettant de contrôler tous les aspects de la transmission de courrier électronique. Les documents joints multiples, l'envoi de copie conforme (*carbon copy*) et de copie discrète (*blind carbon*) ainsi que toute spécification d'en-tête de courrier peuvent être gérés.

### **Comprendre la distribution du courrier**

L'un des concepts clés pour la compréhension du fonctionnement des commandes SMTP se rapporte au mode de distribution du courrier à ses destinataires. Les commandes SMTP ne transmettent pas directement le courrier à chaque destinataire. Elles effectuent la composition et le formatage appropriés au courrier et transmettent le résultat au serveur SMTP spécifié par la commande SMTP\_Host. Le serveur SMTP est généralement une machine se trouvant dans votre entreprise ou chez votre fournisseur d'accès à Internet. Il détermine alors le chemin de transmission optimal pour votre courrier et programme sa distribution en fonction des paramètres configurés par l'administrateur du courrier.

### **Conditions minimales requises pour envoyer un message SMTP complexe**

Pour une bonne transmission des messages électroniques via SMTP, les commandes doivent avoir été correctement paramétrées. La séquence de commandes suivante représente le minimum requis afin que la transmission de courrier puisse aboutir :

- SMTP\_New  
Réserve un espace mémoire pour le nouveau message et fournit une référence devant être utilisée par les commandes ultérieures.
- SMTP\_Host  
Spécifie le serveur SMTP auquel le message doit être transmis.
- SMTP\_From  
Une adresse au moins doit être spécifiée dans l'en-tête "From".
- SMTP\_To  
Une adresse au moins doit être spécifiée dans l'en-tête "To".
- SMTP\_Send  
Envoie le message.
- SMTP\_Clear  
Réinitialise (libère) l'espace mémoire réservé pour le message.

Si seules ces commandes sont exécutées, le message envoyé ne contient ni objet ni corps. Il est nécessaire de définir des informations supplémentaires pour envoyer un message.

---

SMTP\_SetPrefs (retoursLigne; typeCorps; longueurLigne) → Entier

Paramètre	Type		Description
retoursLigne	Entier	→	1 = [défaut] Ajouter, 0 = Ne pas ajouter, -1 = Aucune modification
typeCorps	Entier	→	Body-Content-Type (1 = [défaut] Détection auto, -1 = Aucune modification)
longueurLigne	Entier long	→	Longueur de ligne maximale (0 = [défaut] Détection auto, -1 = Aucune modification)
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_SetPrefs définit les préférences des messages SMTP à envoyer. Elle a une portée globale et interprocess : elle agit sur tous les messages ultérieurs créés avec des commandes SMTP et ce, dans tous les process 4D. Les options paramétrables s'appliquent au message lors de son envoi au serveur SMTP à l'aide des commandes SMTP\_QuickSend ou SMTP\_Send.

A la différence des applications Macintosh qui considèrent un retour chariot seul comme étant un marqueur de fin de ligne ou de paragraphe, les serveurs SMTP requièrent la combinaison de caractères retour chariot/retour à la ligne (CR/LF) pour indiquer la fin d'une ligne.

Le paramètre retoursLigne spécifie comment gérer les retours chariot dans le corps d'un message :

- Si vous passez 0 (zéro), le texte du corps du message est inchangé, ce qui vous permet de gérer vos propres ajouts de retours à la ligne.
- Si vous passez 1 (paramètre par défaut), toutes les combinaisons retour chariot/retour ligne seront automatiquement remplacées par des retours chariot seuls.
- Si vous passez -1, la valeur courante du paramètre est inchangée.

En cas de doute, choisissez 1, la valeur par défaut.

Le paramètre `typeCorps` permet d'indiquer le jeu de caractères utilisé dans le corps du message à envoyer (*Body-Content-Type*) et le type d'encodage à effectuer sur le corps du message (*Content-Transfer-Encoding*), conformément aux valeurs fournies dans le tableau ci-dessous. Si, par exemple, vous passez 2 ("US-ASCII & 7 bits") dans `typeCorps`, vous indiquez que le jeu de caractères utilisé est de l'ASCII US (utilisation des 128 premiers caractères de la table ASCII exclusivement), et 4D IC encodera le message sur 7 bits. A noter que la commande `SMTP_SetPrefs` n'effectue pas de conversion dans le jeu de caractères spécifié, c'est à l'utilisateur de s'assurer de la conformité du jeu de caractères. Si vous souhaitez convertir le jeu de caractères utilisé dans un message, vous devez appeler explicitement la commande `SMTP_CharSet`. Par défaut, ce paramètre a pour valeur 1, les commandes SMTP détectent automatiquement les paramètres appropriés en fonction du contenu du corps du message.

- 1 Aucune modification
- 0 Application & binaire ; pas d'encodage
- 1 Valeur par défaut ; sélection de "US-ASCII & 7 bits" ou "ISO-8859-1 & quortable-printable" en fonction du contenu du message.
- 2 US-ASCII & 7 bits
- 3 US-ASCII & quortable-printable
- 4 US-ASCII & base64
- 5 ISO-8859-1 & quortable-printable
- 6 ISO-8859-1 & base64
- 7 ISO-8859-1 & 8 bits
- 8 ISO-8859-1 & binaire
- 9 Réservé
- 10 ISO-2022-JP (support du japonais) & 7 bits
- 11 ISO-2022-KR (support du coréen) & 7 bits
- 12 ISO-2022-CN (support du chinois traditionnel et simplifié) & 7 bits
- 13 HZ-GB-2312 (support du chinois simplifié) & 7 bits
- 14 Shift-JIS (support du japonais) & base64\*

\* Disponible à compter de la version 6.7 des commandes Internet de 4D.

**Note :** Avec 4D IC version 6.5.1, dans le cas de l'utilisation des valeurs 10 à 13, le message doit avoir été encodé sur 7 bits préalablement à l'appel des commandes `SMTP_QuickSend` et `SMTP_Body`.

Le paramètre `longueurLigne` spécifie la longueur maximale de ligne dans le corps du message. Les commandes SMTP "forcent" le passage à la ligne dans le corps du texte en insérant un retour chariot/retour ligne après le mot le plus proche de la longueur de ligne maximale. Tout nombre peut être spécifié, mais il est préférable que les lignes ne dépassent pas 80 caractères. Passez -1 pour laisser inchangée la valeur courante.



Par défaut, longueurLigne prend la valeur 0 (zéro). Dans ce cas, les commandes SMTP utilisent les valeurs recommandées par les RFC, en fonction du typeCorps. Les passages à la ligne sont alors insérés selon le tableau suivant :

Type de corps	Passage à la ligne à
Base64	76
Quoted-printable	76
Autres...	pas de passage à la ligne

Le traitement du passage à la ligne est fortement recommandé car de nombreux systèmes et programmes de messagerie ne peuvent gérer les messages contenant des lignes de longueur illimitée. En outre, un courrier passe par de nombreux systèmes avant d'atteindre sa destination finale et un ordinateur situé sur le trajet peut rejeter un message s'il est incapable d'en interpréter le format.

### Référence

SMTP\_Charset, SMTP\_GetPrefs.

SMTP\_GetPrefs (retoursLigne; typeCorps; longueurLigne) → Entier

Paramètre	Type	Description
retoursLigne	Entier	← 0 = Ne pas ajouter, 1 = Ajouter RetoursLigne
typeCorps	Entier	← Body-Content-Type (Type de contenu du corps)
longueurLigne	Entier long	← Longueur de ligne maximale
Résultat	Entier	← Code d'erreur

### Description

La commande SMTP\_GetPrefs renvoie les paramètres courants des préférences SMTP. Les valeurs par défaut sont retournées, à moins qu'un appel préalable à SMTP\_SetPrefs ne les ait modifiées. Pour une description complète de ces paramètres, reportez-vous à la commande SMTP\_SetPrefs.

Le paramètre retoursLigne retourne la manière dont les commandes SMTP gèrent les retours chariot dans le corps d'un message.

Le paramètre typeCorps retourne le type courant de contenu du corps (*Body-Content-Type*).

Le paramètre longueurLigne retourne la longueur maximale courante de ligne du texte dans le corps du message.

### Référence

SMTP\_SetPrefs.

SMTP\_QuickSend (nomServeur; msgDe; msgA; objet; message) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur
msgDe	Texte	→	Adresse électronique ou Liste d'adresses
msgA	Texte	→	Adresse électronique ou Liste d'adresses
objet	Texte	→	Objet du message
message	Texte	→	Message
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_QuickSend vous permet de créer et d'envoyer un message en une seule commande. Si vous avez besoin d'un plus grand contrôle sur votre message, ou s'il est plus complexe, utilisez plutôt la commande SMTP\_New.

Le paramètre nomServeur contient le nom ou l'adresse IP du serveur SMTP auquel le message sera envoyé pour distribution.

Le paramètre msgDe contient une ou plusieurs adresses électroniques complètes indiquant l'expéditeur initial du message. Toutes les adresses figurant dans l'en-tête Émetteur (*From*) sont visibles par tous les destinataires du message.

Le paramètre msgA contient une ou plusieurs adresses électroniques complètes. Les adresses figurant dans l'en-tête msgA recevront chacune une copie originale du message. Chaque destinataire du message pourra visualiser les autres adresses électroniques auxquelles le message a été envoyé.

Le paramètre objet contient un texte concis décrivant l'objet du message.

**Attention :** En principe, l'objet et le corps du message ne doivent pas contenir de caractères accentués (tels que é, ö, etc.) lors de leur envoi. Si vous souhaitez malgré tout utiliser des caractères de ce type (appelés "caractères étendus"), reportez-vous aux commandes SMTP\_SetPrefs et SMTP\_Charset.

Le paramètre message contient le corps du message électronique. La taille du message est limitée aux 32 Ko d'une variable ou d'un champ texte 4e Dimension.

## Exemple

Exemple d'utilisation de cette commande :

```
$NomServeur:="www.4d.com"
$MsgÀ:="adupont@4d.fr"
$MsgDe:="jsmith@4d.com"
$Objet:="Rapport de ventes"
$Message:="Pouvez-vous m'envoyer le rapport des ventes de janvier 1999 ? Merci."
⇒ $Erreur:=SMTP_QuickSend ($NomServeur;$MsgDe;$MsgÀ;$Objet;$Message)
  Si ($Erreur#0)
    ALERTE("Erreur: SMTP_QuickSend"+ Caractere(13)+IT_ErrorText ($Erreur))
  Fin de si
```

## Référence

SMTP\_Charset, SMTP\_New, SMTP\_SetPrefs.

SMTP\_New (smtp\_ID) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	←	Référence du nouveau message
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_New doit être la première commande exécutée dans la séquence de création d'un message électronique SMTP, sauf si SMTP\_QuickSend est utilisée. SMTP\_New crée un nouveau message en mémoire et renvoie sa référence sous forme d'entier long dans le paramètre smtp\_ID. Les commandes SMTP ultérieures devront utiliser cette référence pour remplir les informations d'en-tête et de corps du message avant que SMTP\_Send soit appelée.

Chaque appel de SMTP\_New doit avoir un appel correspondant de SMTP\_Clear. Après avoir envoyé un message, l'appel de SMTP\_Clear libère la mémoire occupée par le contenu du message.

Le paramètre smtp\_ID retourne le numéro d'identification du message créé. Cet ID est utilisé pour toutes les références ultérieures à ce message. Il est possible d'ouvrir simultanément plusieurs nouveaux messages, le smtp\_ID retourné pour chacun d'eux fournit un moyen d'identifier le message auquel doit s'appliquer une commande.

### Exemples

Reportez-vous aux exemples des commandes SMTP\_Body et SMTP\_Send.

### Référence

SMTP\_Clear, SMTP\_QuickSend, SMTP\_Send.

SMTP\_Host (smtp\_ID; nomServeur{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence du message
nomServeur	Alpha	→	Nom ou adresse IP du serveur
supprimerOption	Entier	→	0 = Utiliser ou remplacer, 1 = Supprimer
Résultat	Entier	←	Code d'erreur

### Description

Tous les messages créés et envoyés avec les commandes SMTP doivent être dirigés vers un serveur SMTP spécifique. 4D Internet Commands ne distribue pas de courrier directement à chaque destinataire, mais uniquement au serveur SMTP désigné par cette commande. Le serveur SMTP est chargé de la résolution des éventuelles erreurs d'adresse et de la distribution du message.

smtp\_ID est l'identifiant du message électronique créé avec la commande SMTP\_New.

nomServeur contient le nom ou l'adresse IP du serveur SMTP qui gèrera la distribution du message.

supprimerOption vous permet de supprimer ou non la valeur courante du paramètre nomServeur :

- Si vous passez 0 (zéro), le serveur utilisé est celui spécifié par nomServeur.
- Si vous passez 1, la valeur de nomServeur définie pour le message identifié par smtp\_ID est supprimée.
- Si vous omettez ce paramètre, par défaut la valeur 0 est utilisée.

### Exemples

Reportez-vous aux exemples des commandes SMTP\_Body et SMTP\_Send.

### Référence

SMTP\_New.

SMTP\_Send (smtp\_ID) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence du message
Résultat	Entier	←	Code d'erreur

Description

La commande SMTP\_Send envoie le message référencé par smtp\_ID,mais n'efface pas les données de la mémoire.

smtp\_ID est l'identifiant du message électronique créé avec la commande SMTP\_New.

Exemple

Dans cet exemple, un message est créé et les éléments statiques sont définis. Ensuite, pour chaque enregistrement de la table [Personnes], le message est personnalisé et envoyé.

```
$erreur:=SMTP_New ($smtp_id)
$erreur:=SMTP_Host ($smtp_id;"wkrp.com")
$erreur:=SMTP_From ($smtp_id;"herb_tarlick@wkrp.com")
$erreur:=SMTP_ReplyTo ($smtp_id;"bigguy@wkrp.com")
$erreur:=SMTP_Subject ($smtp_id;"Promotions sur les espaces publicitaires !")
Boucle($i;1;Enregistrements trouves ([Personnes]))
  Si ([Personnes]VentesACeJour>100000)
    $Corps:=◇GrdTexteDisque
  Sinon
    $Corps:=◇PttTexteDisque
  Fin de Si
  $Corps:=Remplacer chaine ($TexteConstant;"<Salutations>";[Personnes]Prénom)
  $erreur:=SMTP_To ($smtp_id;[Personnes]Email;1)
    `Remplacer l'en-tête "A" par une nouvelle valeur
  $erreur:=SMTP_Body ($smtp_id;$Corps)
⇒ $erreur:=SMTP_Send ($smtp_id)
  ENREGISTREMENT SUIVANT([Personnes])
Fin de boucle
$erreur:=SMTP_Clear ($smtp_id)
```

Référence

SMTP\_New.

SMTP\_Clear (smtp\_ID) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→ ←	Référence du message 0 si exécution correcte
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_Clear supprime le message désigné par smtp\_ID, libérant la mémoire utilisée lors de sa création. A chaque appel de SMTP\_New doit correspondre un appel à SMTP\_Clear.

smtp\_ID contient l'identifiant du message électronique créé avec la commande SMTP\_New.

Si la commande SMTP\_Clear a été correctement exécutée, smtp\_ID retourne la valeur 0 (zéro).

### Exemples

Reportez-vous aux exemples des commandes SMTP\_Body et SMTP\_Send.

### Référence

SMTP\_New.



SMTP\_Date (smtp\_ID; msgDate; msgHeure; fuseauHoraire; décalage{; supprimerOption}) → Entier

Paramètre	Type	Description
smtp_ID	Entier long →	Référence du message
msgDate	Date →	Date de création du message
msgHeure	Heure →	Heure de création du message
fuseauHoraire	Entier →	Code d'emplacement
décalage	Entier →	Dépend de la valeur du paramètre fuseauHoraire
supprimerOption	Entier →	0 = Ajouter/Remplacer, 1 = Effacer
Résultat	Entier ←	Code d'erreur

### Description

La commande SMTP\_Date crée l'en-tête Date du message désigné par le paramètre smtp\_ID. La date et l'heure fournies à la commande doivent correspondre à l'emplacement courant de la machine envoyant le message. Les paramètres suivants doivent respecter un format spécifique de manière à ce que le serveur de courrier électronique réceptionnant le message puisse déterminer la date et l'heure locales en fonction de la date, l'heure, la zone horaire et le décalage qui lui sont communiqués.

**Note :** Si un message électronique est composé sans en-tête Date, le serveur SMTP en ajoute une en fonction de ses réglages courants d'heure et de date. Tous les messages électroniques SMTP contiennent un en-tête Date ajouté soit par l'application cliente, soit par le serveur SMTP.

Le paramètre smtp\_ID contient l'identifiant du message créé avec la commande SMTP\_New.

Le paramètre msgDate est une date 4D qui indique la date de création du message.

Le paramètre msgHeure indique l'heure de création du message.

Le paramètre fuseauHoraire identifie le fuseau horaire de l'émetteur. Vous pouvez passer toute valeur comprise entre 0 et 6, en fonction des indications suivantes :

- 0 (zéro) permet de spécifier directement dans le paramètre décalage le nombre d'heures à soustraire ou à ajouter au temps universel (TU).

- Si vous passez 1, la machine émettrice ajoutera automatiquement le décalage fondé sur la PRAM du Macintosh. Lorsque vous passez 1 dans fuseauHoraire, le paramètre décalage est inutile. Le fuseau horaire d'un ordinateur Macintosh est déterminé par les réglages du tableau de bord **Planisphère** (MacOS 8) ou **Date et heure** (MacOS 8.5 et suivants). Vous devez veillez à l'exactitude de ce paramétrage si les valeurs horaires sont un facteur primordial de votre base de données.
- Les valeurs de 2 à 5 correspondent aux 4 fuseaux horaires des Etats-Unis. Le décalage pour chacune de ces valeurs spécifie si cette zone horaire est en heure d'été (décalage = 1) ou non (décalage = 0).
- Si vous passez 6, vous indiquez que le temps est défini sur 24 heures. Dans ce cas, le décalage est déterminé par le tableau horaire ci-dessous. Passez la valeur de décalage correspondante (de -12 à 12) au code horaire sur 24 heures de l'emplacement de l'émetteur.

La valeur du paramètre décalage dépend du code défini dans le paramètre fuseauHoraire. Reportez-vous aux descriptions et au tableau horaire suivants pour connaître la valeur correcte à passer dans ce paramètre.

Code	Fuseau horaire	Paramètre de décalage
0	+/- décalage TU	Le décalage est en heures +/-
1	+/- décalage TU	Le décalage n'est pas utilisé, il est fourni par la PRAM du Mac
2	HNE - HAE	( 0 = HNE, 1 = HAE )
3	HNC - HAC	( 0 = HNC, 1 = HAC )
4	HNR - HAR	( 0 = HNR, 1 = HAR )
5	HNP - HAP	( 0 = HNP, 1 = HAP )
6	Heure sur 24 heures	<i>Voir table ci-dessous</i>

Valeurs de décalage	Codes horaires sur 24 heures
0	Z
-1 à -9	A à I
-10 à -12	K à M
1 à 12	N à Y

#### Définitions des abréviations

TU	Temps Universel
HNE	Heure normale de l'Est
HAE	Heure avancée de l'Est
HNC	Heure normale du Centre
HAC	Heure avancée du Centre
HNR	Heure normale des Rocheuses
HAR	Heure avancée des Rocheuses
HNP	Heure normale du Pacifique
HAP	Heure avancée du Pacifique

supprimerOption :

- Passez 0 (zéro) dans ce paramètre pour ajouter un nouvel en-tête Date ou remplacer le précédent.
- Passez 1 pour effacer toute valeur préalablement définie dans cet en-tête.
- Si vous omettez ce paramètre, la valeur 0 est utilisée.

### Référence

SMTP\_New.

SMTP\_From (smtp\_ID; msgEmetteur{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence du message
msgEmetteur	Texte	→	Adresse électronique ou Liste d'adresses
supprimerOption	Entier	→	0 = Ajouter, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

**Description**

La commande SMTP\_From contient la ou les adresse(s) électronique(s) de la ou des personne(s) devant figurer dans l'en-tête "Emetteur(s)" (*From*) du message. Ces personnes sont responsables de la création ou de l'envoi du message. Généralement, l'en-tête "Emetteur(s)" contient l'adresse de la personne qui a composé et envoyé le message. Cependant, lorsque le message est créé par un groupe de personnes, chaque membre doit être individuellement listé dans l'en-tête "Emetteur(s)".

L'en-tête "Emetteur(s)" est obligatoire. Si une adresse figure dans l'en-tête "Emetteur(s)", la présence de l'en-tête "Expéditeur" (*Sender*) est optionnelle.

smtp\_ID contient l'identifiant du message électronique créé avec la commande SMTP\_New.

msgEmetteur peut contenir une ou plusieurs adresses électroniques. Toutes les adresses figurant dans l'en-tête "Emetteur(s)" sont visibles par les destinataires du message.

**Note sur les réponses automatiques :** En l'absence d'en-tête "Réponse à" (*ReplyTo*) dans le message identifié par smtp\_ID, toutes les réponses au message seront adressées à chaque personne figurant dans l'en-tête "Emetteur(s)".

Le paramètre supprimerOption vous permet de préciser s'il faut conserver ou supprimer la zone d'en-tête "Emetteur(s)" éventuellement existante :

- Si vous passez 0 (zéro), le contenu du paramètre passé est ajouté à l'en-tête existant.
- Si vous passez 1, le contenu du paramètre passé remplace le contenu de l'en-tête existant. Dans ce cas, si vous avez passé une chaîne vide dans msgEmetteur, l'en-tête "Emetteur(s)" est supprimé.
- Si vous passez 2, l'en-tête "Emetteur(s)" est supprimé du message.
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.

### Exemple

Dans cet exemple, trois personnes composent un message, concernant une modification de politique de l'entreprise, qui doit être distribué à tout le personnel de la société. Les réponses à ce message seront adressées aux personnes présentes dans l'en-tête "Emetteur(s)".

```
$Emetteur:="prez@acme.com,vp@acme.com,cfo@acme.com"
⇒ $Erreur:=$SMTP_From ($smtp_id;$Emetteur;0)
   $Erreur:=$SMTP_Subject ($smtp_id;"Changement de politique de l'entreprise";0)
   $Erreur:=$SMTP_To ($smtp_id;␣Tous_employés;0)
```

### Référence

SMTP\_New.

SMTP\_Sender (smtp\_ID; msgExpéditeur; supprimerOption) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence du message
msgExpéditeur	Texte	→	Adresse électronique (1 seulement)
supprimerOption	Entier	→	0 = Ajouter, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_Sender permet de paramétrer l'adresse e-mail de la personne qui envoie le message. Cette commande est destinée à être utilisée lorsque l'expéditeur n'est pas le véritable auteur du message, ou pour désigner la personne qui, dans un groupe d'auteurs, a réellement envoyé le message. Cette commande n'est pas nécessaire lorsque le contenu du champ "Expéditeur" (*Sender*) est redondant avec celui du champ "Emetteur(s)" (*From*).

Lorsque les messages sont créés et envoyés automatiquement par un programme, la zone d'en-tête "Expéditeur" doit désigner le compte e-mail de l'administrateur du programme et non le compte géré par le programme.

smtp\_ID contient l'identifiant du message électronique créé avec la commande SMTP\_New.

Le paramètre msgExpéditeur contient l'adresse électronique à insérer dans l'en-tête "Expéditeur" du message. **Une** seule adresse peut être spécifiée pour cet en-tête.

Le paramètre supprimerOption permet de spécifier s'il faut conserver ou supprimer la zone d'en-tête "Expéditeur" éventuellement existante :

- Si vous passez 0 (zéro), le contenu du paramètre passé est ajouté au contenu de l'en-tête existant.
- Si vous passez 1, le contenu du paramètre passé remplace le contenu de l'en-tête existant. Dans ce cas, si vous avez passé une chaîne vide, l'en-tête "Expéditeur" est supprimé.
- Si vous passez 2, l'en-tête "Expéditeur" est supprimé du message.
- Si ce paramètre est omis, par défaut la valeur 0 est utilisée.

## Exemple

Dans cet exemple, trois cadres composent un message concernant une modification de politique de l'entreprise. Ce message doit être distribué à tout le personnel de la société par la secrétaire. Les réponses éventuelles devront être envoyées aux trois personnes répertoriées dans l'en-tête "Emetteur(s)".

```
$Emetteur:="prez@acme.com,vp@acme.com,cfo@acme.com"
$Erreur:=SMTP_From ($smtp_id;$Emetteur;0)
⇒ $Erreur:=SMTP_Sender ($smtp_id;"secretaire@acme.com";0)
  $Erreur:=SMTP_Subject ($smtp_id;"Changement de politique de l'entreprise";0)
  $Erreur:=SMTP_To ($smtp_id;◇Tous_employés;0)
```

## Référence

SMTP\_New.

SMTP\_ReplyTo (smtp\_ID; réponseA{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence du message
réponseA	Texte	→	Adresse électronique ou Liste d'adresses
supprimerOption	Entier	→	0 = Ajouter à la liste existante, 1 = Remplacer les anciennes valeurs par les nouvelles, 2 = Supprimer les adresses spécifiées
Résultat	Entier	←	Code d'erreur

Description

La commande SMTP\_ReplyTo vous permet de prédéfinir le contenu par défaut de la liste d'adresses qui sera utilisée lorsque le récepteur répondra à un message. Normalement, c'est le contenu de l'en-tête "Emetteur(s)" (*From*) qui est utilisé dans ce cas. Toutefois, cet en-tête est ignoré lorsqu'un en-tête "Réponse à" (*ReplyTo*) a été défini pour le message.

Pour le développeur de bases de données, SMTP\_ReplyTo peut être un outil puissant de contrôle des destinataires des réponses dans des messageries automatiques. La commande permet d'envoyer des réponses à des adresses autres que celles inscrites dans les en-têtes "Emetteur(s)" ou "Expéditeur", par exemple à un compte spécifique créé pour assurer le suivi des réponses.

smtp\_ID contient l'identifiant du message électronique par la commande SMTP\_New.

Le paramètre réponseA contient une ou plusieurs adresses électroniques. Les adresses listées dans cet en-tête seront utilisées par le logiciel de messagerie des destinataires comme adresse(s) de réponse par défaut.

Le paramètre supprimerOption vous permet de préciser comment gérer la ou les adresse(s) insérée(s) dans réponseA :

- Si vous passez 0 (zéro), les nouvelles valeurs seront ajoutées à celles éventuellement présentes dans cet en-tête.
- Si vous passez 1, les valeurs éventuellement présentes sont remplacées par les nouvelles valeurs. Si vous avez passé une chaîne vide dans réponseA, vous obtiendrez le même résultat qu'en passant 2.
- Si vous passez 2, l'en-tête ainsi que son contenu sont supprimés.
- Si supprimerOption est omis, la valeur 0 est utilisée par défaut.



## Exemple

Dans cet exemple, trois cadres composent un message concernant un changement de politique de l'entreprise. Ce message doit être adressé à chaque employé de la société par la secrétaire. Les réponses à ce message seront retournées à la secrétaire et au "service\_du\_personnel", et non aux cadres.

```
$Emetteur:="prez@acme.com,vp@acme.com,cfo@acme.com"
$Erreur:=SMTP_From ($smtp_id;$Emetteur;0)
$Erreur:=SMTP_Sender ($smtp_id;"secretaire@acme.com";0)
⇒ $Erreur:=SMTP_ReplyTo ($smtp_id;"secretaire@acme.com,
                                service_du_personnel@acme.com";0)
$Erreur:=SMTP_Subject ($smtp_id;"Changement de politique de l'entreprise";0)
$Erreur:=SMTP_To ($smtp_id;⋄Tous_employés;0)
```

## Référence

SMTP\_New.

SMTP\_To (smtp\_ID; msgA{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
msgA	Texte	→	Adresse électronique ou liste d'adresses
supprimerOption	Entier	→	0 = Ajouter, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

**Description**

La commande SMTP\_To inscrit l'adresse des destinataires principaux d'un message dans la zone d'en-tête "A" (*To*). Toutes les adresses listées dans les zones d'en-tête "A" et "Cc" d'un message électronique seront visibles par chaque destinataire.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

msgA contient une ou plusieurs adresses électroniques.

Le paramètre supprimerOption vous permet de préciser s'il faut conserver ou supprimer la zone d'en-tête "A" éventuellement existante :

- Si vous passez 0 (zéro), le contenu du paramètre passé est ajouté au contenu de l'en-tête existant.
- Si vous passez 1, le contenu du paramètre passé remplace le contenu de l'en-tête existant. Dans ce cas, si vous avez passé une chaîne vide dans msgA, l'en-tête "A" est supprimé.
- Si vous passez 2, l'en-tête "A" est supprimé du message.
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.

**Exemple**

Voir l'exemple de la commande SMTP\_Body.

**Référence**

SMTP\_New.

---

SMTP\_Cc (smtp\_ID; copieConforme{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
copieConforme	Texte	→	Adresse électronique ou Liste d'adresses
supprimerOption	Entier	→	0 = Ajouter, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_Cc ajoute des destinataires en "copie conforme" (*Carbon copy*) au message spécifié par smtp\_ID. Pour envoyer un message, il n'est pas obligatoire que l'en-tête "Cc" contienne des adresses. Toutes les adresses listées dans les zones d'en-tête "A" et "Cc" d'un message électronique seront visibles par chaque destinataire.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

Le paramètre copieConforme contient une ou plusieurs adresses électroniques.

Le paramètre supprimerOption vous permet de préciser s'il faut conserver ou supprimer la zone d'en-tête "Cc" éventuellement existante :

- Si vous passez 0 (zéro), le contenu du paramètre passé est ajouté au contenu de l'en-tête existant.
- Si vous passez 1, le contenu du paramètre passé remplace le contenu de l'en-tête existant. Dans ce cas, si vous avez passé une chaîne vide dans copieConforme, l'en-tête "Cc" est supprimé.
- Si vous passez 2, l'en-tête "Cc" est supprimé du message.
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.

### Exemple

Voir l'exemple de la commande SMTP\_Body.

### Référence

SMTP\_Bcc, SMTP\_New.

SMTP\_Bcc (smtp\_ID; copieDiscrète{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
copieDiscrète	Texte	→	Liste d'adresses
supprimerOption	Entier	→	0 = Ajouter, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_Bcc ajoute des destinataires en "copie discrète" (*Blind carbon copy*) au message spécifié par smtp\_ID. Pour envoyer un message, le champ "Bcc" n'est pas obligatoire.

La seule façon de préserver la confidentialité de la liste d'adresses lors de l'envoi de courrier à un ensemble de personnes est d'insérer ces adresses dans la zone d'en-tête "Bcc". Les adresses listées dans cet en-tête ne sont pas envoyées dans le corps ni dans l'en-tête du message. Ces adresses ne peuvent être lues par aucun destinataire du message.

Un destinataire "Bcc" peut visualiser tous les destinataires "A" et "Cc", mais pas les autres destinataires "Bcc". Généralement, lors d'envoi de courrier en masse, tous les destinataires doivent être placés dans l'en-tête "Bcc". Cela permet d'éviter que les messages reçus soient encombrés par une grande liste d'adresses, et de ne pas communiquer les adresses des destinataires.

Une autre raison de l'utilisation de "Bcc" est que la plupart des messageries électroniques disposent d'une fonction "Répondre à tous". Celle-ci transfère toutes les adresses présentes dans les champs "A" et "Cc" du message reçu dans le champ "A" du message à renvoyer. Placer les adresses dans l'en-tête "Bcc" permet d'éviter qu'un ou plusieurs destinataires répondent à tous les destinataires initiaux.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

Le paramètre copieDiscrète contient une ou plusieurs adresses.

Le paramètre `supprimerOption` vous permet de préciser s'il faut conserver ou supprimer la zone d'en-tête "Cc" éventuellement existante :

- Si vous passez 0 (zéro), le contenu du paramètre passé est ajouté au contenu de l'en-tête existant.
- Si vous passez 1, le contenu du paramètre passé remplace le contenu de l'en-tête existant. Dans ce cas, si vous avez passé une chaîne vide dans `copieDiscrète`, l'en-tête "Bcc" est supprimé.
- Si vous passez 2, l'en-tête "Bcc" est supprimé du message.
- Si `supprimerOption` est omis, par défaut la valeur 0 est utilisée.

### Exemple

Dans cet exemple, pour chaque enregistrement de la table [Personnes], une adresse est ajoutée à la liste de copies discrètes :

```
$erreur:=SMTP_From ($smtp_id;"sales@massmarket.com")
$erreur:=SMTP_Subject ($smtp_id;"Ventes incroyables ! Seulement cette semaine !")
$erreur:=SMTP_Body ($smtp_id;$CorpsGénérique)
Boucle($i;1;Enregistrements_trouves([Personnes]))
⇒ $erreur:=SMTP_Bcc ($smtp_id;[Personnes]Email;0)
   `Ajoute cette adresse e-mail à la liste BCC
   ENREGISTREMENT SUIVANT([Personnes])
Fin de boucle
$erreur:=SMTP_Send ($smtp_id) `Envoie le message à tout le monde
$erreur:=SMTP_Clear ($smtp_id)
```

### Référence

SMTP\_Cc, SMTP\_New.

SMTP\_InReplyTo (smtp\_ID; msgEnRéponseA{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence du message
msgEnRéponseA	Texte	→	Texte en réponse à
supprimerOption	Entier	→	0 = Remplacer sauf si msgEnReponseA vide, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

Description

La commande SMTP\_InReplyTo insère, lors de la réponse à un message, le texte du message d'origine.

smtp\_ID contient l'identifiant d'un message électronique créé avec SMTP\_New.

Le paramètre msgEnRéponseA contient le texte des messages précédents auxquels se rapporte ce message. Pour toute référence se rapportant à son formatage, veuillez consulter la RFC 822.

**Attention :** Le texte ne doit pas contenir de retours à la ligne (code ASCII=10). Un retour à la ligne désigne la fin de la section d'en-tête et le début du corps du texte. Les en-têtes suivants risquent alors d'être considérés comme le corps du texte et de ne pas être correctement reconnus par le logiciel du serveur ou du client. Pour plus d'informations sur les zones d'en-tête, veuillez consulter la RFC 822.

Le paramètre optionnel supprimerOption vous permet de préciser s'il faut conserver ou supprimer l'en-tête "Réponse A" éventuellement existant :

- Si vous passez 0 (zéro), le contenu du paramètre msgEnRéponseA remplace dans la zone d'en-tête tout texte éventuellement présent, sauf si vous passez une chaîne vide dans msgEnRéponseA, auquel cas l'en-tête "Réponse A" existant et son contenu sont conservés.
  - Si vous passez 1, le contenu du paramètre msgEnRéponseA remplace dans la zone d'en-tête tout texte éventuellement présent. Dans ce cas, si vous passez une chaîne vide dans msgEnRéponseA, l'en-tête "Réponse A" existant est supprimé.
  - Si vous passez 2, l'en-tête "Réponse A" est supprimé du message.
- Si supprimerOption est omis, la valeur 0 est utilisée par défaut.

Référence

SMTP\_New.

SMTP\_References (smtp\_ID; msgRéférences{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
msgRéférences	Texte	→	Texte de référence
supprimerOption	Entier	→	0 = Remplacer sauf si msgRéférences vide, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

Description

La commande SMTP\_References insère, lors de la réponse à un message, des références supplémentaires au texte d'origine.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

Le paramètre msgRéférences contient le texte des références supplémentaires. Pour des informations se rapportant à son formatage, veuillez consulter la RFC 822.

**Attention :** Le texte ne doit pas contenir de retours à la ligne (code ASCII=10). Un retour à la ligne désigne la fin de la section d'en-tête et le début du corps du texte. Les en-têtes suivants risquent alors d'être considérés comme le corps du texte et de ne pas être correctement reconnus par le logiciel du serveur ou du client. Pour plus d'informations sur les zones d'en-tête, veuillez consulter la RFC 822.

Le paramètre optionnel supprimerOption vous permet de préciser s'il faut conserver ou supprimer l'en-tête "Références" éventuellement existant :

- Si vous passez 0 (zéro), le contenu du paramètre msgRéférences remplace dans la zone d'en-tête tout texte éventuellement présent, sauf si vous passez une chaîne vide dans msgRéférences, auquel cas l'en-tête "Références" existant et son contenu sont conservés.
  - Si vous passez 1, le contenu du paramètre msgRéférences remplace dans la zone d'en-tête tout texte éventuellement présent. Dans ce cas, si vous passez une chaîne vide dans msgRéférences, l'en-tête "Références" existant est supprimé.
  - Si vous passez 2, l'en-tête "Références" est supprimé du message.
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.

Référence

SMTP\_New.

SMTP\_Comments (smtp\_ID; msgCommentaires{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
msgCommentaires	Texte	→	Texte du commentaire
supprimerOption	Entier	→	0 = Remplacer sauf si msgCommentaires vide, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

Description

La commande SMTP\_Comments permet d'ajouter des commentaires au message sans modifier le corps du message. Les commentaires n'apparaissent que dans la zone d'en-tête. De nombreux logiciels de messagerie n'affichent pas l'intégralité des en-têtes du message.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

Le paramètre msgCommentaires contient le texte que vous souhaitez placer en tant que commentaire dans l'en-tête du message.

**Attention :** Le texte ne doit pas contenir de retours à la ligne (code ASCII=10). Un retour à la ligne désigne la fin de la section d'en-tête et le début du corps du texte. Les en-têtes suivants risquent alors d'être considérés comme le corps du texte et de ne pas être correctement reconnus par le logiciel du serveur ou du client. Pour plus d'informations sur les zones d'en-tête, veuillez consulter la RFC 822.

Le paramètre optionnel supprimerOption vous permet de préciser s'il faut conserver ou supprimer l'en-tête "Commentaires" éventuellement existant :

- Si vous passez 0 (zéro), le contenu du paramètre msgCommentaires remplace dans la zone d'en-tête tout texte éventuellement présent, sauf si vous passez une chaîne vide dans msgCommentaires, auquel cas l'en-tête "Commentaires" existant et son contenu sont conservés.
  - Si vous passez 1, le contenu du paramètre msgCommentaires remplace dans la zone d'en-tête tout texte éventuellement présent. Dans ce cas, si vous passez une chaîne vide dans msgCommentaires, l'en-tête "Commentaires" existant est supprimé.
  - Si vous passez 2, l'en-tête "Commentaires" est supprimé du message.
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.



**Exemple**

Reportez-vous à l'exemple de la commande SMTP\_Body.

**Référence**

SMTP\_New.

SMTP\_Keywords (smtp\_ID; msgMotsClés; supprimerOption)) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
msgMotsClés	Texte	→	Liste de mots-clés
supprimerOption	Entier	→	0 = Remplacer sauf si msgMotsClés vide, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

Description

Le commande SMTP\_Keywords permet d'insérer des mots-clés dans l'en-tête "Mots-clés" (*Keywords*) du message désigné par smtp\_ID.

smtp\_ID est l'identifiant d'un message électronique créé avec la commande SMTP\_New.

Le paramètre msgMotsClés contient un ou plusieurs mots-clés, séparés par des virgules. Pour plus d'informations sur le formatage de ce paramètre, veuillez consulter la RFC 822.

**Attention :** Le texte ne doit pas contenir de retours à la ligne (code ASCII=10). Un retour à la ligne désigne la fin de la section d'en-tête et le début du corps du texte. Les en-têtes suivants risquent alors d'être considérés comme le corps du texte et de ne pas être correctement reconnus par le logiciel du serveur ou du client. Pour plus d'informations sur les zones d'en-tête, veuillez consulter la RFC 822.

Le paramètre optionnel supprimerOption vous permet de préciser s'il faut conserver ou supprimer l'en-tête "Mots-clés" éventuellement existant :

- Si vous passez 0 (zéro), le contenu du paramètre msgMotsClés remplace dans la zone d'en-tête tout texte éventuellement présent, sauf si vous passez une chaîne vide dans msgMotsClés, auquel cas l'en-tête "Mots-clés" existant et son contenu sont conservés.
  - Si vous passez 1, le contenu du paramètre msgMotsClés remplace dans la zone d'en-tête tout texte éventuellement présent. Dans ce cas, si vous passez une chaîne vide dans msgMotsClés, l'en-tête "Mots-clés" existant est supprimé.
  - Si vous passez 2, l'en-tête "Mots-clés" est supprimé du message.
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.

Référence

SMTP\_New.

---

SMTP\_Encrypted (smtp\_ID; msgCrypté{; supprimerOption})

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence du message
msgCrypté	Texte	→	Type d'encryptage
supprimerOption	Entier	→	0 = Remplacer sauf si msgCrypté vide, 1 = Remplacer, 2 = Supprimer

### Description

La commande SMTP\_Encrypted permet d'informer les utilisateurs du type d'encryptage utilisé dans le corps du message. Les commandes Internet de 4D **ne permettent pas** d'encrypter ou de décrypter des messages électroniques. Le cryptage du corps du message relève de la responsabilité du développeur. Si le corps du message est encrypté (avant son affectation via SMTP\_Body), il est nécessaire d'utiliser cette commande pour indiquer au logiciel de messagerie le type d'encryptage employé.

smtp\_ID contient l'identifiant d'un message électronique créé avec SMTP\_New.

Le paramètre msgCrypté indique le type d'encryptage utilisé. L'en-tête "Encryptage" est utilisé par le logiciel de messagerie du destinataire pour décrypter du corps du message reçu. Pour plus amples informations sur le formatage, veuillez consulter la RFC 822.

**Attention :** Le texte ne doit pas contenir de retours à la ligne (code ASCII=10). Un retour à la ligne désigne la fin de la section d'en-tête et le début du corps du texte. Les en-têtes suivants risquent alors d'être considérés comme le corps du texte et de ne pas être correctement reconnus par le logiciel du serveur ou du client. Pour plus d'informations sur les zones d'en-tête, veuillez consulter la RFC 822.

Le paramètre optionnel supprimerOption vous permet de préciser s'il faut conserver ou supprimer l'en-tête "Encryptage" éventuellement existant :

- Si vous passez 0 (zéro), le contenu du paramètre msgCrypté remplace dans la zone d'en-tête tout texte éventuellement présent, sauf si vous passez une chaîne vide dans msgCrypté, auquel cas l'en-tête "Encryptage" existant et son contenu sont conservés.
  - Si vous passez 1, le contenu du paramètre msgCrypté remplace dans la zone d'en-tête tout texte éventuellement présent. Dans ce cas, si vous passez une chaîne vide dans msgCrypté, l'en-tête "Encryptage" existant est supprimé.
  - Si vous passez 2, l'en-tête "Encryptage" est supprimé du message.
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.

### Référence

SMTP\_Body, SMTP\_New.

SMTP\_AddHeader (smtp\_ID; nomEnTête; texteEnTête{; supprimerOption}) → Entier

Paramètre	Type	Description
smtp_ID	Entier long →	Référence de message
nomEnTête	Alpha →	Nom de l'en-tête
texteEnTête	Texte →	Texte de l'en-tête
supprimerOption	Entier →	0 = Ajouter 1 = Remplacer tous les en-têtes par 'nomEnTête', 2 = Supprimer tous les en-têtes nommés 'nomEnTête'
Résultat	Entier ←	Code d'erreur

Description

La commande SMTP\_AddHeader vous permet d'ajouter votre propre en-tête au message référencé par smtp\_ID. Outre les en-têtes standard directement gérés par les commandes Internet de 4D, il existe deux catégories d'en-têtes supplémentaires : les en-têtes "utilisateur" (*User-defined*) et les en-têtes "étendus" (*Extended*). La commande SMTP\_AddHeader permet d'ajouter à la fois un nouvel en-tête et les données qu'il contient.

**En-têtes "étendus" :** Ces en-têtes ont été officiellement reconnus par le NIC et ont été définis après les spécifications SMTP initiales. Ces en-têtes ont généralement un rôle spécifique affectant le comportement de différents logiciels. Les en-têtes "étendus" ne commencent jamais par la lettre "X".

**En-têtes "utilisateur" :** Le protocole SMTP permet à quiconque de créer ses propres définitions d'en-tête. Tous les en-têtes définis par l'utilisateur doivent commencer par les caractères "X-" pour éviter tout conflit avec un futur en-tête "étendu". Les en-têtes "utilisateur" sont particulièrement utiles lorsque vous contrôlez les deux extrémités des communications.

Les en-têtes "utilisateur" permettent au développeur de stocker des données qui peuvent être facilement extraites au moyen de la commande MSG\_FindHeader. Par exemple, vous pouvez créer un en-tête nommé "X-001001", qui contient la valeur du champ 01 de la table 01. Il est possible d'ajouter un nombre illimité d'en-têtes à un message. Les en-têtes "utilisateur" permettent d'insérer des informations plus faciles à extraire que celles nécessitant l'analyse du corps du message.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

Le paramètre nomEnTête contient le nom de l'en-tête à ajouter.

Le paramètre texteEnTête contient les informations à affecter à la zone d'en-tête identifiée par nomEnTête.

**Attention :** Le texte ne doit pas contenir de retours à la ligne (code ASCII=10). Un retour à la ligne désigne la fin de la section d'en-tête et le début du corps du texte. Les en-têtes suivants risquent alors d'être considérés comme le corps du texte et de ne pas être correctement reconnus par le logiciel du serveur ou du client. Pour plus d'informations sur les zones d'en-tête, veuillez consulter la RFC#822.

Le paramètre supprimerOption vous permet de préciser s'il faut ou non supprimer l'en-tête courant.

- Si vous passez 0 (zéro), l'en-tête nomEnTête est ajouté au message.
- Si vous passez 1, tous les en-têtes du message sont remplacés par l'en-tête nomEnTête. Dans ce cas, si nomEnTête contient une chaîne vide, tous les en-têtes du message sont supprimés.
- Si vous passez 2, tous les en-têtes nomEnTête sont supprimés du message.

### Exemple

Si vous souhaitez envoyer un mail en HTML, le corps du message doit contenir les balises HTML (ex : <HTML>, <HEAD>, etc.) et les en-têtes "Content-Type" doivent être remplacés par "text/html; charset=us-ascii" :

```
Si(Sous chaine($corps;1;6)="<HTML>")  
⇒      $err:=SMTP_AddHeader($SMTP_ID;"Content-Type: "; "text/html; charset=us-ascii";1)  
Fin de si
```

### Référence

MSG\_FindHeader, SMTP\_New.

SMTP\_Subject (smtp\_ID; msgObjet{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
msgObjet	Texte	→	Objet du message
supprimerOption	Entier	→	0 = Remplacer sauf si msgObjet vide, 1 = Remplacer, 2 = Supprimer
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_Subject ajoute l'en-tête "Objet" (*Subject*) au message référencé par smtp\_ID. Si un objet avait déjà été défini par une précédente commande SMTP\_Subject, le nouvel objet écrase le précédent.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

msgObjet contient un texte concis décrivant le sujet traité en détail dans le corps du message.

### Attention :

- L'en-tête "Objet" ne doit pas contenir, en principe, de caractères accentués (tels que é, ö, etc.). Si vous souhaitez malgré tout utiliser des caractères de ce type (appelés "caractères étendus"), reportez-vous aux commandes SMTP\_SetPrefs et SMTP\_Charset.
- Le texte ne doit pas contenir de retours à la ligne (code ASCII=10). Un retour à la ligne désigne la fin de la section d'en-tête et le début du corps du texte. Les en-têtes suivants risquent alors d'être considérés comme le corps du texte et de ne pas être correctement reconnus par le logiciel du serveur ou du client. Pour plus d'informations sur les zones d'en-tête, veuillez consulter la RFC 822.

Le paramètre optionnel `supprimerOption` vous permet de préciser s'il faut conserver ou supprimer l'en-tête "Objet" éventuellement existant :

- Si vous passez 0 (zéro), le contenu du paramètre `msgObjet` remplace dans la zone d'en-tête tout texte éventuellement présent, sauf si vous passez une chaîne vide dans `msgObjet`, auquel cas l'en-tête "Objet" existant et son contenu sont conservés.
  - Si vous passez 1, le contenu du paramètre `msgObjet` remplace dans la zone d'en-tête tout texte éventuellement présent. Dans ce cas, si vous passez une chaîne vide dans `msgObjet`, l'en-tête "Objet" existant est supprimé.
  - Si vous passez 2, l'en-tête "Objet" est supprimé du message.
- Si `supprimerOption` est omis, par défaut la valeur 0 est utilisée.

### **Exemple**

Reportez-vous à l'exemple de la commande `SMTP_Body`.

### **Référence**

`SMTP_Charset`, `SMTP_New`, `SMTP_SetPrefs`.

SMTP\_Body (smtp\_ID; msgCorps{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
msgCorps	Texte	→	Corps du message
supprimerOption	Entier	→	0 = Remplacer sauf si msgCorps vide, 1 = Remplacer, 2 = Ajouter
Résultat	Entier	←	Code d'erreur

### Description

La commande SMTP\_Body insère le texte de msgCorps dans le corps principal du message identifié par smtp\_ID. Le msgCorps est le principal bloc de texte.

smtp\_ID contient l'identifiant d'un message électronique créé avec la commande SMTP\_New.

Le paramètre msgCorps contient le corps du message. La taille de msgCorps est limitée aux 32 Ko d'un objet 4D de type Texte. Cela ne signifie pas que le message lui-même est limité à 32 Ko : pour envoyer une lettre dont le corps est supérieur à 32 Ko, il vous suffit de concaténer (mettre bout à bout) plusieurs textes, à l'aide du paramètre supprimerOption (voir plus bas). La taille du corps d'un message n'est réellement limitée que par la mémoire disponible.

**Attention :** En principe, le corps du message ne doit pas contenir de caractères accentués (tels que é, ö, etc.). Si vous souhaitez malgré tout utiliser des caractères de ce type (appelés “caractères étendus”), reportez-vous aux commandes SMTP\_SetPrefs et SMTP\_Charset.

Le paramètre optionnel supprimerOption vous permet de préciser s'il faut conserver ou remplacer le corps du message éventuellement existant :

- Si vous passez 0 (zéro), le contenu du paramètre msgCorps remplace le corps de message éventuellement présent, sauf si vous passez une chaîne vide dans msgCorps, auquel cas le corps de message existant est conservé.
  - Si vous passez 1, le contenu du paramètre msgCorps remplace le corps de message éventuellement présent. Dans ce cas, si vous passez une chaîne vide dans msgCorps, le corps du message est supprimé du message.
  - Si vous passez 2, le contenu de msgCorps est ajouté au corps du message, à la suite de tout texte déjà défini (concaténation).
- Si supprimerOption est omis, par défaut la valeur 0 est utilisée.



## Exemple

Voici un exemple SMTP complet :

```
C_ENTIER LONG($SMTP_ID)
C_BOOLEEN($EnvoyeOK;$OK)
$EnvoyeOK:=Faux `Indicateur précisant s'il est applicable à toutes les commandes
Au cas ou
  : (Non(VérifErreur ("SMTP_New";SMTP_New ($SMTP_ID))))
  : (Non(VérifErreur ("SMTP_Host";SMTP_Host ($SMTP_ID;␣pref_Server))))
  : (Non(VérifErreur ("SMTP_From";SMTP_From ($SMTP_ID;vDe))))
  : (Non(VérifErreur ("SMTP_To";SMTP_To ($SMTP_ID;vA))))
  : (Non(VérifErreur ("SMTP_Cc";SMTP_Cc ($SMTP_ID;vCC))))
  : (Non(VérifErreur ("SMTP_Bcc";SMTP_Bcc ($SMTP_ID;vBcc))))
  : (Non(VérifErreur ("SMTP_Subject";SMTP_Subject ($SMTP_ID;vSujet))))
  : (Non(VérifErreur ("SMTP_Comments";SMTP_Comments ($SMTP_ID;
    "Envoyé via 4D"))))
  : (Non(VérifErreur ("SMTP_AddHeader";SMTP_AddHeader ($SMTP_ID;
    "X-4Ddemo:";␣VERSION))))
⇒  : (Non(VérifErreur ("SMTP_Body";SMTP_Body ($SMTP_ID;vMessage))))
    : (Non(VérifErreur ("SMTP_Send";SMTP_Send ($SMTP_ID))))
Sinon
  $EnvoyeOK:=Vrai `Message composé et envoyé avec succès
Fin de cas

Si ($SMTP_ID#0) `Si un message a été créé en mémoire, il faut l'effacer maintenant
  $OK:=VérifErreur ("SMTP_Clear";SMTP_Clear ($SMTP_ID))
Fin de si
```

L'exemple suivant fournit le code de la méthode *VérifErreur*. Cette méthode reçoit deux paramètres : le nom de la commande (\$Commande) et la valeur de l'erreur (fournie par l'exécution de la commande dans le paramètre de la méthode). *VérifErreur* renvoie une valeur booléenne indiquant si la commande a retourné le code d'erreur 0 (zéro). Si ce n'est pas le cas, la valeur retournée (\$0) est Faux, sinon Vrai.

```
C_TEXTE(vMsgErreur)
$Commande:=$1
$Erreur:=$2
$Resultat:=Vrai
Si ($Erreur#0)
    $Resultat:=Faux
    Si (◇VOIRERREURS) `Booléen pour déterminer s'il faut afficher les messages d'erreur
        vMsgErreur:=IT_ErrorText ($Erreur)
        ALERTE("ERREUR ---"+Caractere(13)+"Commande : "+$Commande+
            Caractere(13)+"Erreur Code:"+Chaine($Erreur)+
            Caractere(13)+"Description : "+vMsgErreur)
    Fin de si
Fin de si
$0:=$Resultat
```

## Référence

SMTP\_Charset, SMTP\_New, SMTP\_SetPrefs.

SMTP\_Attachment (smtp\_ID; nomFichier; typeEncodage{; supprimerOption}) → Entier

Paramètre	Type		Description
smtp_ID	Entier long	→	Référence de message
nomFichier	Texte	→	Nom du fichier à joindre
typeEncodage	Entier	→	0 = Pas d'encodage (n'envoie que la DataFork) ±1 = BinHex ±2 = Base64 (n'envoie que la DataFork) ±3 = AppleSingle ±4 = AppleDouble ±5 = AppleSingle ET Base64 ±6 = AppleDouble ET Base64 ±7 = UUEncode
supprimerOption	Entier	→	0 = Ajouter à la liste existante, 1 = Remplacer toutes les pièces jointes par nomFichier, 2 = Ne supprimer que cette pièce jointe
Résultat	Entier	←	Code d'erreur

**Description**

La commande SMTP\_Attachment vous permet de joindre des fichiers de type binaire ou texte à votre message dans le format MIME. Cette commande peut être appelée plusieurs fois pour joindre plusieurs documents à un message. Si vous passez une valeur positive dans le paramètre typeEncodage, l'encodage sera effectué au moment de l'envoi du message.

smtp\_ID contient l'identifiant d'un message créé avec la commande SMTP\_New.

Le paramètre nomFichier désigne le fichier que vous voulez joindre au message. Cette valeur peut être spécifiée de trois manières :

- " " = Affiche la boîte de dialogue standard d'ouverture de documents.
- "nomFichier" = Recherche le nom du fichier dans le répertoire de la structure de la base de données.
- "Chemin:nomFichier" = Chemin d'accès complet (nom du fichier compris).

Le paramètre `typeEncodage` indique le type d'encodage à appliquer au fichier avant de l'intégrer au message. S'il s'agit d'un fichier binaire, vous devez utiliser un type d'encodage approprié (BinHex, AppleSingle). L'encodage le plus courant est BinHex.

Si la valeur de `typeEncodage` est positive, la commande encode automatiquement le fichier au moment de l'envoi du message à l'aide de la commande `SMTP_Send`.

Si le fichier est volumineux, l'exécution de la commande `SMTP_Send` peut durer quelques instants. Il est possible de gagner du temps lorsqu'un même fichier est envoyé plusieurs fois : l'astuce consiste à encoder le fichier une seule fois avec la commande `IT_Encode`, puis de le joindre au message en passant une valeur négative dans `typeEncodage`. Lorsque le paramètre `typeEncodage` reçoit une valeur négative, aucun encodage supplémentaire n'est effectué, mais le type d'encodage est décrit dans l'en-tête du message du fichier joint, de manière à ce que le logiciel de messagerie du destinataire puisse l'interpréter correctement.

**Note :** Vous ne devez pas passer d'élément de tableau dans le paramètre `typeEncodage`.

Le paramètre `supprimerOption` indique comment traiter le fichier joint.

- Si vous passez 0 (zéro), le fichier joint est ajouté à la liste courante des fichiers joints.
- Si vous passez 1, `nomFichier` remplace tous les fichiers éventuellement déjà joints au message. Dans ce cas, si `nomFichier` est une chaîne vide, tous les fichiers joints sont supprimés.
- Si vous passez 2, le fichier désigné par `nomFichier` est supprimé de la liste des fichiers joints.

## Référence

`IT_Encode`, `SMTP_New`, `SMTP_Send`.

SMTP\_Charset (encoderEntêtes; jeuCorps) → Entier

Paramètre	Type		Description
encoderEntêtes	Entier	→	-1 = Utiliser le paramétrage courant, 0 = Ne rien faire, 1 = Convertir dans le jeu de caractères spécifié si ISO-8859-1 ou ISO-2022-JP, encoder les caractères étendus
jeuCorps	Entier	→	-1 = Utiliser le paramétrage courant, 0 = Ne rien faire, 1 = Convertir dans le jeu de caractères spécifié si ISO-8859-1 ou ISO-2022-JP
Résultat	Entier	←	Code d'erreur

Description

La commande SMTP\_Charset automatise le support des caractères étendus dans les messages lors de leur envoi. Si cette commande n'est pas appelée ou si ses deux paramètres sont mis à 0, les commandes Internet de 4D version 6.7 ou supérieure fonctionneront de la même manière qu'en version 6.5.x.

La commande SMTP\_Charset permet, d'une part, d'indiquer si le jeu de caractères défini dans le paramètre typeCorps de la commande SMTP\_Prefs doit être appliqué aux en-têtes et au corps des messages à envoyer ; d'autre part, elle permet de définir si un en-tête comportant des caractères étendus doit être encodé sous la forme “=?ISO-8859-1?Q?Test=E9?= ...”, conformément au RFC 1342.

Cette commande a une portée globale et interprocess : elle agit sur tous les messages ultérieurs envoyés à l'aide des commandes SMTP\_QuickSend et SMTP\_Send et ce, dans tous les process 4D.

La commande SMTP\_Charset est particulièrement utile pour le traitement des caractères étendus dans les en-têtes “Subject” ou les noms insérés dans les adresses (par exemple, pour l'encodage d'adresses sous la forme “=?ISO-8859-1?Q?Test=E9?= <test@n.net>”).

En fonction des champs d'en-tête, l'encodage (toujours en base 64 et dans le jeu de caractères fixé par SMTP\_SetPrefs) est effectué de la manière suivante :

- Pour les en-têtes “Subject” et “Comments” (en-têtes appelés “non structurés”) : toute la chaîne est encodée si elle comporte des caractères étendus ;
- Pour les en-têtes “From”, “To”, “Cc”, “Bcc”, “Sender”, “ReplyTo”, “InReplyTo” (en-têtes appelés “structurés”) :
  - Ce qui est entre < > est systématiquement considéré comme une adresse eMail et n'est jamais encodé ;
  - Les caractères séparateurs tels que SPC < > ( ) @ , ; : \ " / ? . = ne sont jamais encodés ;
  - Ce qui se trouve entre deux séparateurs est encodé s'il y a des caractères étendus.

Exemples d'adresses :

- someone@somewhere n'est pas encodé ;
- Michèle <michele@somewhere>, seul le mot Michèle est encodé.

Le paramètre `encoderEntêtes` définit les traitements à appliquer aux champs d'en-tête lors de l'envoi des messages. Par défaut, ce paramètre a pour valeur 0.

- -1 : Utiliser les paramétrages courants ;
- 0 : Ne rien faire
- 1 : - D'une part, si la commande `SMTP_SetPrefs` définit le jeu de caractères comme étant de l'ISO-8859-1 ou de l'ISO-2022-JP, l'en-tête est converti dans le jeu de caractères spécifié,  
- D'autre part, les champs d'en-tête sont encodés sous la forme "=?Jeu de caractères spécifié par SMTP\_SetPrefs?Encodage Base64 systématiquement?Test=E9?=..." (RFC 1342) s'ils contiennent des caractères étendus et ce, quel que soit le jeu de caractères spécifié.

**Note :** Les en-têtes de type `X_Mailer` doivent être en ASCII US.

Le paramètre `jeuCorps` définit les traitements à appliquer au corps du message lors de son envoi. Par défaut, ce paramètre a pour valeur 0.

- -1 : Utiliser les paramétrages courants ;
- 0 : Ne rien faire ;
- 1 : Si la commande `SMTP_SetPrefs` définit le jeu de caractères comme étant de l'ISO-8859-1 ou de l'ISO-2022-JP, le texte du corps du message est converti dans le jeu de caractères spécifié.

## Exemple

Exemple de traitement des caractères étendus :

```
SMTP_SetPrefs(1;1;0)
⇒ $err:=SMTP_Charset(1;1)
   $err:=SMTP_QuickSend("monmail.com";"monadresse";"destination";"L'euro ¤";
                        "Le symbole de l'Euro est ¤")
   `Le sujet et le corps du message sont convertis en ISO-8859-1,
   `Le sujet est encodé conformément au RFC 1342
```

## Référence

POP3\_Charset, SMTP\_SetPrefs.

# 3

---

## IT Review Mail





Les commandes POP3 permettent à votre base de données de récupérer des messages d'un serveur de courrier POP3. Les commandes Internet de 4D sont conformes aux spécifications MIME, elles peuvent reconnaître et extraire des messages contenant plusieurs pièces jointes.

Les commandes POP3 sont réparties en deux thèmes, "IT Review Mail" (préfixées "POP3") et "IT Downloaded Mail" (préfixées "MSG"), correspondant aux deux modes de lecture du courrier électronique. Le premier mode consiste à prendre connaissance du contenu du courrier, à le télécharger. Le second mode consiste à travailler sur les messages téléchargés.

La taille des fichiers à télécharger va déterminer l'utilisation d'un mode par rapport à l'autre. Par exemple, un seul message électronique auquel est joint un fichier de 5 Mo pourrait facilement dépasser la capacité de stockage de la base de données. Seul un BLOB ou une image 4D est capable d'accueillir des données de cette taille, mais la conversion d'un message ou d'un document joint dans ce format est souvent inefficace car la messagerie cliente doit mobiliser de grandes ressources mémoire pour accéder à l'image ou au BLOB.

Pour résoudre ce problème, la commande POP3\_Download transfère un message du serveur POP3 directement sur le disque local de l'utilisateur. Il suffit ensuite d'utiliser les commandes du thème "IT Downloaded Mail" pour manipuler le fichier sur disque.

L'utilisation des commandes POP3 nécessite une bonne compréhension des paramètres numéroMsg et uniqueID. numéroMsg représente le numéro d'un message dans la boîte aux lettres au moment de l'exécution de la commande POP3\_Login. A la connexion, les messages de la boîte aux lettres sont numérotés de 1 à x (x étant le nombre d'éléments présents dans la boîte aux lettres). Les numéros sont affectés en fonction de l'ordre dans lequel les messages ont été reçus, le numéro 1 étant le plus ancien. Les numéros affectés aux messages ne sont valides que pendant la période comprise entre POP3\_Login et POP3\_Logout.

Au moment de l'exécution de POP3\_Logout, tout message marqué comme "devant être supprimé" disparaît. Lorsque l'utilisateur se reconnecte au serveur, les messages présents dans la boîte aux lettres sont de nouveau numérotés de 1 à x. Par exemple, s'il y a 10 messages dans la boîte aux lettres, et si les messages numérotés de 1 à 5 sont supprimés, les messages 6 à 10 seront renumérotés de 1 à 5 la prochaine fois que l'utilisateur consultera sa boîte aux lettres.

Pour illustrer ce fonctionnement, supposons que vous vous connectiez à un serveur POP3 et obteniez la liste de messages suivante :

<b>n°Msg</b>	<b>uniqueID</b>	<b>Date</b>	<b>De</b>	<b>Objet</b>
1	bd573a4dbd573a4d	1 Jul 1998...	jimw@acme.com	Clients potentiels ...
2	bd574dc7bd574dc7	1 Jul 1998...	frank@acme.com	Commande de licence sur site
3	bd575f06bd575f06	3 Jul 1998...	joe@acme.com	Qui veut déjeuner ?
4	bd5761d4bd5761d4	4 Jul 1998...	kelly@acme.com	Appel de votre femme...
5	bd577dc7db577dc5	4 Jul 1995...	track@fedex.com	Suivi FedEx

Pendant la session, vous supprimez les messages 3 et 4. Lorsque vous quittez la session, vos demandes de suppression sont exécutées. Lorsque vous retournez sur le serveur, la liste de messages est alors renumérotée ainsi :

<b>n°Msg</b>	<b>uniqueID</b>	<b>Date</b>	<b>De</b>	<b>Objet</b>
1	bd573a4dbd573a4d	1 Jul 1998...	jimw@acme.com	Clients potentiels ...
2	bd574dc7bd574dc7	1 Jul 1998...	frank@acme.com	Commande de licence sur site
3	bd577dc7db577dc5	4 Jul 1995...	track@fedex.com	Suivi FedEx

numéroMsg n'est pas une valeur statique se rapportant à un message spécifique, elle indique la position relative d'un message de la boîte aux lettres au moment de l'ouverture de la session.

En revanche, uniqueID est un numéro unique affecté au message lors de sa réception par le serveur. Ce numéro est défini par le serveur POP3 sur la base de l'heure et de la date auxquelles le message est reçu. Malheureusement, les serveurs POP3 n'utilisent pas uniqueID comme référence principale des messages. Aussi, lorsque vous manipulez des messages avec les commandes POP3, vous devez passer numéroMsg comme paramètre d'identification des messages sur le serveur. Par conséquent, vous devez être prudent lorsque vous développez des applications qui référencent des messages dans la base de données tout en laissant le contenu du message sur le serveur.

Cette section est consacrée aux commandes permettant de travailler avec les messages disponibles sur le serveur POP3 (thème "IT Review Mail").

---

POP3\_SetPrefs (retoursLigne; dossierMsg; dossierDocsJoint) → Entier

Paramètre	Type		Description
retoursLigne	Entier	→	0 = Ne pas retirer les retours à la ligne, 1 = Retirer les retours à la ligne, -1 = Aucune modification
dossierMsg	Texte	→	Chemin d'accès au dossier des messages ("" = aucune modification)
dossierDocsJoint	Texte	→	Chemin d'accès du dossier des documents joint ("" = aucune modification)
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_SetPrefs définit des préférences générales pour toutes les commandes POP3 ultérieures.

Le paramètre `retoursLigne` vous permet de préciser comment traiter les caractères de retour à la ligne dans les messages enregistrés. La plupart des serveurs POP3 associent un caractère Retour chariot (*carriage return*) et un caractère Retour à la ligne (*line feed*) pour indiquer la fin d'une ligne, à la différence des applications Macintosh qui requièrent un simple retour chariot. Dans ce cas, cette option vous permet de supprimer les caractères Retour à la ligne superflus du texte des messages.

- Si vous passez 0, les messages récupérés seront conservés dans le format du serveur POP3.
- Si vous passez 1, les caractères Retour à la ligne seront supprimés des messages récupérés.
- Si vous passez -1, l'option reste telle qu'elle était précédemment définie.

Par défaut, cette option a pour valeur 1, les retours à la ligne rencontrés dans les messages sont automatiquement supprimés.

Le paramètre `dossierMsg` indique le chemin d'accès local du dossier dans lequel les messages récupérés à l'aide de la commande POP3\_Download doivent être enregistrés par défaut.

Le paramètre `dossierDocsJoint` indique le chemin d'accès local du dossier dans lequel les fichiers joints doivent être enregistrés lorsque la commande MSG\_Extract extrait les documents joints du corps du message.

### Référence

MSG\_Extract, POP3\_Download, POP3\_GetPrefs.

POP3\_GetPrefs (retoursLigne; dossierMsg; dossierDocsJoint) → Entier

Paramètre	Type		Description
retoursLigne	Entier	←	0 = Ne pas retirer les RC/RL, 1 = Retirer les RC/RL
dossierMsg	Texte	←	Chemin d'accès au dossier des messages ("" = aucune modification)
dossierDocsJoint	Texte	←	Chemin d'accès du dossier des documents joint ("" = aucune modification)
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_GetPrefs vous permet de connaître les préférences courantes pour les commandes POP3.

Le paramètre `retoursLigne` retourne le paramétrage courant de l'option de suppression des retours à la ligne.

Le paramètre `dossierMsg` retourne le chemin d'accès local du dossier dans lequel sont enregistrés par défaut les messages récupérés.

Le paramètre `dossierDocsJoint` retourne le chemin d'accès local du dossier dans lequel sont enregistrés par défaut les documents joints extraits des messages.

### Référence

POP3\_SetPrefs.

---

POP3\_Login (nomServeur; nomUtilisateur; motDePasse; aPOP; pop3\_ID) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur de courrier POP3
nomUtilisateur	Alpha	→	Nom de l'utilisateur
motDePasse	Alpha	→	Mot de passe
aPOP	Entier	→	0 = Connexion texte clair, 1 = Connexion APOP
pop3_ID	Entier long	←	Référence à cette connexion POP3
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_Login connecte l'utilisateur défini par nomUtilisateur et motDePasse au serveur de courrier POP3 nomServeur. Si aPOP vaut 1, le mécanisme APOP (cf. RFC 1321) est utilisé pour la connexion. Si aPOP vaut zéro ou est vide, la connexion est effectuée normalement, avec un mot de passe en clair. POP3\_Login retourne un numéro d'identification pour la connexion (pop3\_ID) que les commandes ultérieures devront utiliser.

**Attention :** Les accès aux serveurs POP3 ne sont pas interactifs. Une fois connecté à un serveur, vous devez effectuer les actions nécessaires, puis vous déconnecter dès que possible. Un serveur POP3 déconnecte automatiquement les sessions qui restent inactives pendant un certain laps de temps. Si l'on se réfère à la RFC pour POP3, la période d'inactivité autorisée est d'environ 30 minutes. Cependant, notre expérience montre que la plupart des serveurs déconnectent les clients inactifs après un délai beaucoup plus court.

Chaque commande en interaction avec le serveur POP3 provoque la réinitialisation du compteur d'inactivité. Si le serveur interrompt votre connexion avant que vous ayez appelé POP3\_Logout, toutes les suppressions que vous avez demandées sont annulées.

Le paramètre nomServeur contient le nom ou l'adresse IP du serveur de courrier POP3. Il est généralement conseillé d'utiliser le nom du serveur.

Le paramètre nomUtilisateur contient le nom de l'utilisateur du serveur de courrier POP3. Ce nom ne doit pas contenir le nom du domaine. Par exemple, dans le cas de l'adresse "jack@4d.com", le nomUtilisateur est "jack".

Le paramètre `motDePasse` est le mot de passe correspondant au `nomUtilisateur` sur le serveur de courrier POP3.

Le paramètre `aPOP` vous permet d'indiquer si le mécanisme APOP doit être utilisé pour la connexion. Passez 1 pour utiliser ce mécanisme APOP, ou 0 (zéro) pour effectuer la connexion par mot de passe en clair. La valeur par défaut est 0.

Le paramètre `pop3_ID` retourne le numéro de référence de la session. Ce numéro sera utilisé par toutes les commandes ultérieures effectuant des actions au sein de cette session.

### **Référence**

`POP3_Logout`.

POP3\_VerifyID (pop3\_ID) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→ ←	Référence à une connexion POP3 0 = La connexion est déjà close
Résultat	Entier	←	Code d'erreur

### Description

Un serveur POP3 déconnecte automatiquement les comptes demeurés inactifs pendant une certaine période, déterminée par son administrateur (ce mécanisme est appelé *timeout*).

Chaque commande en interaction avec le serveur POP3 provoque la remise à zéro du compteur d'inactivité.

La commande POP3\_VerifyID provoque également la remise à zéro du compteur d'inactivité de la session POP3 spécifiée. Elle permet de conserver une session active en cas de risque de dépassement du délai.

Lors de son exécution, la commande POP3\_VerifyID vérifie que la connexion n'a pas déjà été close.

Si la session est toujours ouverte, la commande indique au serveur POP3 de remettre à zéro le compteur d'inactivité pour la session.

Si la connexion a déjà été fermée, POP3\_VerifyID retourne l'erreur appropriée, libère la mémoire utilisée par la session POP3 et renvoie 0 dans le paramètre pop3\_ID.

Le paramètre pop3\_ID contient une référence de session ouverte créée avec POP3\_Login.

### Référence

POP3\_Login.

POP3\_Reset (pop3\_ID) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence d'une connexion POP3
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_Reset permet d'annuler l'option "à supprimer" appliquée aux messages durant la session courante.

**Note :** La commande POP3\_Delete ne supprime pas immédiatement les messages, son action consiste uniquement à marquer les messages comme étant "à supprimer". Les messages ne sont réellement supprimés sur le serveur POP3 que si la déconnexion (POP3\_Logout) a été correctement effectuée.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

### Référence

POP3\_Delete, POP3\_Login.



POP3\_Delete (pop3\_ID; premierMsg; dernierMsg) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence à une connexion POP3
premierMsg	Entier long	→	Numéro du premier message
dernierMsg	Entier long	→	Numéro du dernier message

Résultat	Entier	←	Code d'erreur
----------	--------	---	---------------

### Description

Dans l'intervalle de messages compris entre premierMsg et dernierMsg, la commande POP3\_Delete marque chaque message comme étant "à supprimer". La suppression réelle des messages ne sera effective que lorsque la commande POP3\_Logout sera correctement exécutée. Si la session courante s'interrompt pour une raison quelconque (délai dépassé, panne de réseau, etc.) avant l'appel à la commande POP3\_Logout, les messages marqués ne seront pas supprimés sur le serveur POP3.

Le paramètre pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

Le paramètre premierMsg désigne le numéro du premier message à supprimer.

Le paramètre dernierMsg désigne le numéro du dernier message à supprimer.

**Note** : Si le paramètre premierMsg est supérieur au paramètre dernierMsg, cette commande ne fait rien. Aucune erreur n'est renvoyée.

### Référence

POP3\_Logout.

POP3\_Logout (pop3\_ID) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→ ←	Référence d'une connexion POP3 0 = Déconnexion réussie
Résultat	Entier	←	Code d'erreur

**Description**

La commande POP3\_Logout referme la session POP3 désignée par pop3\_ID. Si la déconnexion s'est correctement déroulée, la valeur 0 (zéro) est retournée dans le paramètre pop3\_ID.

Lorsque vous vous déconnectez d'un serveur POP3, les éventuelles suppressions demandées pendant la session sont effectuées sur le serveur. Si vous voulez annuler ces suppressions, appelez la commande POP3\_Reset avant POP3\_Logout.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

**Référence**

POP3\_Reset.

POP3\_BoxInfo (pop3\_ID; nombreMsg; tailleMsg) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence d'une connexion POP3
nombreMsg	Entier long	←	Nombre de messages
tailleMsg	Entier long	←	Taille des messages
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_BoxInfo renvoie le nombre et la taille des messages présents dans la boîte aux lettres de la session référencée par pop3\_ID.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

nombreMsg retourne le nombre de messages présents dans la boîte aux lettres.

tailleMsg retourne la taille totale des messages présents dans la boîte aux lettres.

### Référence

POP3\_Login.

POP3\_MsgInfo (pop3\_ID; numéroMsg; tailleMsg; uniqueID) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence d'une connexion POP3
numéroMsg	Entier long	→	Numéro du message
tailleMsg	Entier long	←	Taille du message
uniqueID	Alpha	←	ID unique d'un message sur le serveur
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_MsgInfo retourne la taille et l'ID unique du message désigné par numéroMsg dans la boîte aux lettres ouverte référencée par pop3\_ID.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

Le paramètre numéroMsg indique le message sur lequel vous souhaitez obtenir des informations. Le numéroMsg représente la position du message dans la liste courante des messages. Attention, le numéroMsg d'un message n'est pas une valeur stable, il diffère d'une session à l'autre en fonction des suppressions effectuées.

Le paramètre tailleMsg retourne la taille du message.

Le paramètre uniqueID retourne l'ID unique du message sur le serveur. L'uniqueID est une valeur affectée au message par le serveur POP3. Cette valeur ne varie pas d'une session à l'autre, à la différence de numéroMsg. La valeur uniqueID est une référence utile pour vérifier si la base de données a déjà téléchargé un message depuis le serveur.

### Référence

POP3\_Login.

POP3\_GetMessage (pop3\_ID; numéroMsg; décalage; longueur; texteMsg) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence d'une connexion POP3
numéroMsg	Entier long	→	Numéro du message
décalage	Entier long	→	Caractère à partir duquel commencer la récupération
longueur	Entier long	→	Nombre de caractères à renvoyer
texteMsg	Texte	←	Texte du message
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_GetMessage retourne le contenu du message désigné par numéroMsg dans la boîte aux lettres référencée par pop3\_ID. Sauf spécification contraire de la commande POP3\_SetPrefs, les caractères Retour à la ligne (*Line feed*) à l'intérieur du message sont supprimés. La commande POP3\_GetMessage renvoie l'intégralité du message, y compris les informations des zones d'en-tête.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

Le paramètre numéroMsg désigne le message à récupérer dans la boîte aux lettres. Le numéroMsg représente la position du message dans la liste courante de messages. Attention, le numéroMsg d'un message n'est pas une valeur stable, il diffère d'une session à l'autre en fonction des suppressions effectuées.

Le paramètre décalage vous permet d'indiquer la position du caractère (calculée par rapport au début du message) à partir duquel commencer la lecture. Dans la plupart des cas, vous passerez 0 (zéro) dans ce paramètre.

Le paramètre longueur indique le nombre de caractères à récupérer au-delà de la position de décalage. La longueur maximale d'une variable 4D de type Texte étant limitée à 32 000 caractères, le paramètre longueur doit être inférieur à 32 000. Les messages dont la taille est supérieure doivent être récupérés sur disque au moyen de la commande POP3\_Download.

Le texte récupéré est retourné dans la variable texteMsg.

### Référence

POP3\_Download, POP3\_SetPrefs.

---

POP3\_MsgLstInfo (pop3\_ID; premierMsg; dernierMsg; tabTaillesMsg; tabNumMsg; tabIDMsg) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence d'une connexion POP3
premierMsg	Entier long	→	Numéro du premier message
dernierMsg	Entier long	→	Numéro du dernier message
tabTaillesMsg	Tab Entier long	←	Tableau des tailles
tabNumMsg	Tab Entier long	←	Tableau des numéros de message
tabIDMsg	Tab Alpha   Txt	←	Tableau des ID uniques
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_MsgLstInfo retourne la taille, le numéro et l'ID unique d'un ensemble de messages contenus dans la boîte aux lettres. Ces informations sont renvoyées dans trois tableaux, chaque élément de tableau correspondant à un message.

La commande POP3\_MsgLstInfo ajuste la taille de chaque tableau au nombre de messages récupérés. Vous devez toutefois déclarer les tableaux avant d'appeler la commande.

La commande POP3\_MsgLstInfo ne renvoie pas d'erreur si elle ne parvient pas à récupérer des informations sur un message particulier. Dans ce cas, la commande ignore le message et ne crée pas d'élément correspondant dans les tableaux. Par conséquent, si la commande réussit à lire tous les messages, le tableau tabNumMsg contiendra une suite continue de valeurs numériques séquentielles. En cas de problème, des numéros seront manquants dans la suite numérique de tabNumMsg.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

Le paramètre premierMsg indique le numéro du premier message à examiner.

Le paramètre dernierMsg indique le numéro du dernier message à examiner.

Ce numéro représente la position d'un message dans la liste des messages de la boîte aux lettres identifiée par pop3\_ID.

Le tableau tabTaillesMsg reçoit la taille de chaque message dont le numéro est compris entre premierMsg et dernierMsg.

Le tableau tabNumMsg reçoit les numéros de chaque message.

Le tableau tabIDMsg reçoit les ID uniques de chaque message.

**Note :** Si le paramètre premierMsg est supérieur au paramètre dernierMsg, cette commande ne fait rien. Aucune erreur n'est renvoyée.

### **Référence**

POP3\_MsgInfo, POP3\_MsgLst.

POP3\_MsgLst (pop3\_ID; premierMsg; dernierMsg; tabEnTêtesMsg; tabNumMsg; tabIDMsg; tabValeursMsg) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence d'une connexion POP3
premierMsg	Entier long	→	Numéro du premier message
dernierMsg	Entier long	→	Numéro du dernier message
tabEnTêtesMsg	Tab Alpha   Txt	→	Tableau des en-têtes à récupérer
tabNumMsg	Tab Entier long	←	Tableau des numéros de message
tabIDMsg	Tab Alpha	←	Tableau alphanumérique des ID uniques
tabValeursMsg	Tab Alpha 2DITxt	←	Tableau 2D des valeurs des en-têtes
Résultat	Entier	←	Code d'erreur

Description

La commande POP3\_MsgLst permet d'obtenir des informations contenues dans les en-têtes d'un ensemble de messages. Vous remplissez le tableau alphanumérique ou texte tabEnTêtesMsg avec les noms des en-têtes que vous souhaitez récupérer. Le tableau 2D tabValeursMsg reçoit le contenu des en-têtes. A chaque en-tête demandé correspond une "ligne" du tableau tabValeursMsg.

La commande POP3\_MsgLst ne peut renvoyer que le contenu des en-têtes, elle ne peut pas servir à récupérer le corps d'un message.

**Note :** Les champs d'en-tête étant susceptibles de contenir des caractères étendus, vous pouvez automatiser la gestion de ceux-ci à l'aide de la commande POP3\_Charset.

Exemple

```
aEnTêtes{1}:="Date:"
aEnTêtes{2}:="From:"
aEnTêtes{3}:="Subject:"
⇒ POP3_MsgLst (⋄POP3_ID; vPremier; vDernier; aEnTêtes; aNumMsg; aUID; aValeurs)
aValeurs{1}{1} ` par exemple "Jeudi 19 novembre 1998, 00:24:02 -0800"
aValeurs{2}{1} ` par exemple "Jack@4d.com"
aValeurs{3}{1} ` par exemple "Appelez votre femme"
```



Les erreurs sont gérées de la façon suivante :

1) Seules les erreurs relatives à la communication sont renvoyées. Si la commande ne peut pas achever sa tâche en raison d'une erreur (réseau, syntaxe, serveur, etc.), le code d'erreur approprié est renvoyé.

2) Si un message appartenant à l'intervalle spécifié n'existe pas ou comporte une erreur :

- Aucun élément de tableau n'est créé pour ce message.
- Aucune erreur n'est renvoyée.

3) L'incapacité à localiser un ou plusieurs en-têtes dans un message ne constitue pas une erreur :

- Un élément de tableau est créé pour le message.
- Les éléments de tableau "numéro" et "ID" contiennent les valeurs appropriées.
- Pour chaque en-tête introuvable dans le message, une chaîne vide ("") est renvoyée à l'élément de tableau.
- Aucun code d'erreur n'est renvoyé.

**Note :** Si le paramètre `premierMsg` est supérieur au paramètre `dernierMsg`, cette commande ne fait rien. Aucune erreur n'est renvoyée.

### Référence

`POP3_Charset`, `POP3_MsgInfo`, `POP3_MsgLstInfo`.

---

POP3\_Download (pop3\_ID; numéroMsg; enTêteSeul; nomFichier) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence d'une connexion POP3
numéroMsg	Entier long	→	Numéro du message
enTêteSeul	Entier	→	0 = Message entier, 1 = En-tête seul
nomFichier	Texte	→	Nom de fichier local
		←	Nom de fichier local utilisé
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_Download permet de télécharger un message d'un serveur POP3 en local sur disque. Tout message POP3 contenant des fichiers joints ou dont la taille est supérieure à 32 Ko devra être téléchargé avec cette commande. Les fichiers joints peuvent être extraits uniquement à partir d'un message préalablement téléchargé.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

Le paramètre numéroMsg désigne le numéro du message à récupérer dans la boîte aux lettres. Ce numéro représente la position d'un message dans la liste courante des messages. Attention, le numéroMsg d'un message n'est pas une valeur stable, il diffère d'une session à l'autre en fonction des suppressions effectuées.

Le paramètre enTêteSeul vous permet de spécifier si vous souhaitez récupérer la totalité du message ou uniquement les informations des en-têtes.

Le paramètre nomFichier désigne le nom et/ou l'emplacement du fichier dans lequel vous souhaitez enregistrer le message. Cette valeur peut être spécifiée de trois manières :

"" = Enregistre le fichier dans le dossier défini par POP3\_SetPrefs, avec le nom "temp1" (si un fichier de ce nom existe déjà, les noms "temp2", "temp3", etc., sont essayés).

"nomFichier" = Enregistre le fichier dans le dossier défini par POP3\_SetPrefs, avec le nom nomFichier.

"Chemin:nomFichier" = Enregistre le fichier en utilisant le chemin spécifié par nomFichier.

Dans les deux premiers cas, en l'absence de dossier spécifié par POP3\_SetPrefs, le message est enregistré dans le dossier de la structure de la base de données (avec 4D monoposte) ou dans le dossier de 4D Client (avec 4D Server).

Après l'exécution de la commande, le nom final du fichier est retourné dans le paramètre nomFichier. Si vous tentez d'appeler POP3\_Download avec un nomFichier qui existe déjà dans le dossier de téléchargement, ce nom est incrémenté et le nom réellement enregistré sur disque est retourné.

### **Référence**

POP3\_SetPrefs.

POP3\_UIDToNum (pop3\_ID; uniqueID; numéroMsg) → Entier

Paramètre	Type		Description
pop3_ID	Entier long	→	Référence à une connexion POP3
uniqueID	Alpha	→	ID unique d'un message sur le serveur
numéroMsg	Entier long	←	Numéro du message
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_UIDToNum retourne le numéro de message numéroMsg correspondant au message désigné par uniqueID dans la boîte aux lettres référencée par pop3\_ID.

Le numéroMsg d'un message étant une valeur fluctuante, relative aux autres messages de la liste, cette commande renvoie la position courante d'un message dont les informations peuvent avoir été récupérées à l'occasion d'une précédente session POP3.

pop3\_ID contient la référence d'une session ouverte avec POP3\_Login.

Le paramètre uniqueID contient l'ID unique du message à localiser sur le serveur POP3. La commande recherche cette valeur dans les en-têtes des messages référencés dans la session identifiée par pop3\_ID. Une fois trouvé, le numéro du message est retourné dans numéroMsg.

Le paramètre numéroMsg retourne le numéro courant du message désigné par uniqueID. Si l'uniqueID ne peut pas être trouvé sur le serveur, numéroMsg retourne 0 (zéro), aucune erreur n'est renvoyée.

POP3\_Charset (decoderEntêtes; jeuCorps) → Entier

Paramètre	Type		Description
decoderEntêtes	Entier	→	-1 = Utiliser le paramétrage courant, 0 = Ne rien faire, 1 = Convertir dans le jeu de caractères MacOS si ISO-8859-1 ou ISO-2022-JP, décoder les caractères étendus
jeuCorps	Entier	→	-1 = Utiliser le paramétrage courant, 0 = Ne rien faire, 1 = Convertir dans le jeu de caractères MacOS si ISO-8859-1 ou ISO-2022-JP
Résultat	Entier	←	Code d'erreur

### Description

La commande POP3\_Charset automatise le traitement des caractères étendus dans les messages lors de leur exploitation via certaines commandes POP3 et MSG. Si cette commande n'est pas appelée ou si ses deux paramètres sont mis à 0, les commandes Internet de 4D version 6.7 ou supérieure fonctionneront de la même manière qu'en version 6.5.x.

La commande POP3\_Charset permet de définir, d'une part, si les en-têtes comportant des caractères étendus doivent être décodés et, d'autre part, si le jeu de caractères utilisé dans le corps des messages et dans les en-têtes doit être converti.

Cette commande est particulièrement utile pour le traitement des caractères étendus dans les en-têtes tels que "Subject" et les noms placés dans les adresses (par exemple, pour le décodage d'adresses sous la forme =?ISO-8859-1?Q?Test=E9?= <test@n.net>).

Le paramètre decoderEntêtes définit les traitements à appliquer aux champs d'en-tête lors de l'exécution des commandes POP3\_MsgLst et MSG\_FindHeader. Par défaut, ce paramètre a pour valeur 0.

- -1 : Utiliser les paramétrages courants ;
- 0 : Ne rien faire ;
- 1 : L'en-tête est décodé si nécessaire. Si l'en-tête est décodé et si le jeu de caractères spécifié est de l'ISO-8859-1 ou de l'ISO-2022-JP, il est converti, respectivement en ASCII MacOS ou en Shift-JIS.

Le paramètre `jeuCorps` définit les traitements à appliquer au corps du message lors de l'exécution de la commande `MSG_GetBody`. Par défaut, ce paramètre a pour valeur 0.

- -1 : Utiliser les paramétrages courants ;
- 0 : Ne rien faire ;
- 1 : Si le jeu de caractères spécifié dans le champ "*Body-Content-Type*" est de l' ISO-8859-1 ou de l'ISO-2022-JP, le texte du corps du message est converti, respectivement en ASCII MacOS ou en Shift-JIS.

## Exemples

(1) Avec les commandes Internet de 4D version 6.5.x :

```
$Err:=MSG_FindHeader($fichMsg;"From";$from)
$from:=ISO vers Mac($from)
$Err:=MSG_FindHeader($fichMsg;"To";$to)
$to:=ISO vers Mac($to)
$Err:=MSG_FindHeader($fichMsg;"Cc";$cc)
$cc:=ISO vers Mac($cc)
$Err:=MSG_FindHeader($fichMsg;"Subject";$subject)
$subject:=ISO vers Mac($subject)

$Err:=MSG_MessageSize($fichMsg;$tailleEntete;$tailleCorps;$tailleMsg)
$Err:=MSG_GetBody($fichMsg;0;$tailleCorps;$Corps)
$Corps:=ISO vers Mac($Corps)
```

(2) Avec les commandes Internet de 4D version 6.7 :

```
⇒ $Err:=POP3_Charset(1;1)
$Err:=MSG_FindHeader($fichMsg;"From";$from)
$Err:=MSG_FindHeader($fichMsg;"To";$to)
$Err:=MSG_FindHeader($fichMsg;"Cc";$cc)
$Err:=MSG_FindHeader($fichMsg;"Subject";$subject)

$Err:=MSG_MessageSize($fichMess;$tailleEntete;$tailleCorps;$tailleMsg)
$Err:=MSG_GetBody($fichMess;0;$tailleCorps;$Corps)
```

## Référence

SMTP\_Charset.

# 4

---

## IT Downloaded Mail





Les commandes POP3 préfixées "MSG\_" permettent de manipuler des messages qui ont été enregistrés sous forme de fichiers locaux au moyen de la commande POP3\_Download (thème "IT Review Mail"). Les commandes Internet de 4D étant entièrement compatibles MIME, vous pourrez aisément extraire et/ou décoder les fichiers joints. Pour plus d'informations sur les normes MIME, veuillez consulter les documents RFC 1521, RFC 1522 et RFC 2045.

Une fois les messages téléchargés et enregistrés localement, les commandes de ce thème fournissent différentes fonctions permettant de manipuler ces documents. Vous pouvez ainsi obtenir des informations sur les composantes du message, séparer l'en-tête du corps du message, détecter et extraire les fichiers joints, ou encore supprimer des documents.

MSG\_FindHeader (nomFichier; libelléEnTête; valeurEnTête) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Nom de fichier
libelléEnTête	Alpha	→	Libellé de l'en-tête ("From:", "To:", "Subject:", etc.)
valeurEnTête	Texte	←	Valeur
Résultat	Entier	←	Code d'erreur

### Description

La commande MSG\_FindHeader recherche le libelléEnTête dans la section d'en-tête de nomFichier et retourne la valeur du champ dans valeurEnTête. nomFichier contient le nom et/ou le chemin d'accès d'un message téléchargé localement par la commande POP3\_Download.

nomFichier désigne le nom ou le chemin d'accès complet du fichier duquel extraire la valeur d'en-tête. Si vous passez un nom de fichier seul, le chemin d'accès par défaut sera celui du dossier défini par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le chemin par défaut sera celui du dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou du dossier de 4D Client (avec 4D Server).

Passez dans libelléEnTête une chaîne de caractères contenant le libellé d'en-tête à rechercher. libelléEnTête peut faire référence à tout en-tête défini, spécifié par l'utilisateur ou étendu, tel que "From:" (Emetteur), "To:" (Destinataire), "X-MonEnTête", etc.

**Note :** Les libellés des en-têtes sont toujours exprimés en anglais.

Le paramètre valeurEnTête reçoit la valeur affectée à la zone d'en-tête spécifiée.

**Note :** Le paramètre valeurEnTête étant susceptible de contenir des caractères étendus, vous pouvez automatiser la gestion de ceux-ci à l'aide de la commande POP3\_Charset.

### Référence

POP3\_Charset, POP3\_Download, POP3\_SetPrefs.

---

MSG\_MessageSize (nomFichier; tailleEnTête; tailleCorps; tailleMsg) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Nom de fichier
tailleEnTête	Entier long	←	Taille de la section d'en-tête (soustrait les retours à la ligne si Prefs ON)
tailleCorps	Entier long	←	Taille du corps (soustrait les retours à la ligne si Prefs ON)
tailleMsg	Entier long	←	Message entier ou taille du fichier (ignore les Prefs)
Résultat	Entier	←	Code d'erreur

### Description

La commande MSG\_MessageSize renvoie des informations sur la taille des différentes parties du message désigné par nomFichier. nomFichier contient le nom et/ou le chemin d'accès d'un message téléchargé localement par la commande POP3\_Download.

nomFichier désigne le nom ou le chemin d'accès complet du fichier duquel extraire les informations du message. Si vous passez un nom de fichier seul, le chemin d'accès par défaut sera celui du dossier défini par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le chemin par défaut sera celui du dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou du dossier de 4D Client (avec 4D Server).

Le paramètre tailleEnTête retourne la taille de la section d'en-tête.

Le paramètre tailleCorps retourne la taille du corps du texte.

Le paramètre tailleMsg retourne la taille globale du message.

### Référence

POP3\_Download, POP3\_SetPrefs.

---

MSG\_GetHeaders (nomFichier; décalage; longueur; texteEnTête) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Nom de fichier
décalage	Entier long	→	Début du décalage dans la section d'en-têtes (0 = début de l'en-tête)
longueur	Entier long	→	Nombre de caractères
texteEnTête	Texte	←	Texte de la section d'en-têtes (supprime les retours à la ligne si Prefs ON)
Résultat	Entier	←	Code d'erreur

### Description

La commande MSG\_GetHeaders renvoie sous forme de texte brut la section d'en-têtes du message désigné par nomFichier. L'en-tête d'un message POP3 comprend tout le texte situé à partir du début du message, jusqu'à la première occurrence de deux séquences consécutives de retour chariot/retour ligne (*carriage return/line feed*).

nomFichier désigne le nom ou le chemin d'accès complet du fichier duquel extraire l'en-tête. Si vous passez un nom de fichier seul, le chemin d'accès par défaut sera celui du dossier défini par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le chemin par défaut sera celui du dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou du dossier de 4D Client (avec 4D Server).

Le paramètre décalage vous permet de définir, dans la section d'en-tête source, la position du caractère à partir duquel commencer la récupération.

Le paramètre longueur définit le nombre de caractères à renvoyer. La longueur de la section d'en-têtes peut être récupérée via MSG\_MessageSize.

Le paramètre texteEnTête retourne le texte brut de l'en-tête.

### Référence

MSG\_MessageSize, POP3\_SetPrefs.

---

MSG\_GetBody (nomFichier; décalage; longueur; texteCorps) → Entier long

Paramètre	Type		Description
nomFichier	Texte	→	Nom de fichier
décalage	Entier long	→	Début du décalage dans le corps du message (0 = début du corps)
longueur	Entier long	→	Nombre de caractères
texteCorps	Texte	←	Texte du corps (supprime les retours à la ligne si Prefs ON)
Résultat	Entier long	←	Code d'erreur

### Description

La commande MSG\_GetBody renvoie uniquement le texte du corps du message désigné par nomFichier. Elle n'inclut pas le texte des pièces jointes et supprime tous les en-têtes MIME.

nomFichier désigne le nom ou le chemin d'accès complet du fichier duquel extraire le corps du message. Si vous passez un nom de fichier seul, le chemin d'accès par défaut sera celui du dossier défini par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le chemin par défaut sera celui du dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou du dossier de 4D Client (avec 4D Server).

Le paramètre décalage vous permet de définir, dans le corps, la position du caractère à partir duquel commencer la récupération.

Le paramètre longueur indique le nombre de caractères à renvoyer.

Le paramètre texteCorps retourne le texte du corps du message.

**Note :** Le paramètre texteCorps étant susceptible de contenir des caractères étendus, vous pouvez automatiser la gestion de ceux-ci à l'aide de la commande POP3\_Charset.

### Référence

POP3\_Charset, POP3\_SetPrefs.

MSG\_GetMessage (nomFichier; décalage; longueur; texteBrut) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Nom de fichier
décalage	Entier long	→	Début du décalage dans le fichier du message (0 = début du fichier)
longueur	Entier long	→	Nombre de caractères
texteBrut	Texte	←	Texte brut (ignore les Prefs)
Résultat	Entier	←	Code d'erreur

### Description

La commande MSG\_GetMessage renvoie le texte brut du message désigné par nomFichier, quelles que soient les pièces jointes. Elle ne supprime pas les en-têtes MIME.

nomFichier désigne le nom ou le chemin d'accès complet du fichier duquel extraire le corps du message. Si vous passez un nom de fichier seul, le chemin d'accès par défaut sera celui du dossier défini par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le chemin par défaut sera celui du dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou du dossier de 4D Client (avec 4D Server).

Le paramètre décalage vous permet de définir, dans le message source, la position du caractère à partir duquel commencer la récupération.

Le paramètre longueur indique le nombre de caractères à renvoyer.

Le paramètre texteBrut retourne l'intégralité du texte du message. Les paramètres des préférences pour la suppression des retours à la ligne (*line feed*) sont ignorés et les éventuels documents joints imbriqués dans le corps du message ne sont pas supprimés.

### Référence

POP3\_SetPrefs.

MSG\_HasAttach (nomFichier; nbreDocsJoint) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Nom de fichier
nbreDocsJoint	Entier	←	Nombre de documents joints
Résultat	Entier	←	Code d'erreur

### Description

La commande MSG\_HasAttach retourne dans le paramètre nbreDocsJoint le nombre de documents joints au message désigné par nomFichier. Un document joint est une pièce jointe qui n'est pas en texte MIME. Si le message n'a pas de document joint, la commande retourne 0 dans nbreDocsJoint.

nomFichier désigne le nom ou le chemin d'accès complet du fichier dans lequel vérifier la présence de documents joints. Si vous passez un nom de fichier seul, le chemin d'accès par défaut sera celui du dossier défini par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le chemin par défaut sera celui du dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou du dossier de 4D Client (avec 4D Server).

Le paramètre nbreDocsJoint retourne le nombre de documents joints à nomFichier.

### Référence

POP3\_SetPrefs.

---

MSG\_Extract (nomFichier; decoder; cheminDocsJoint; listePiècesJointes) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Nom de fichier
decoder	Entier	→	0 = Pas de décodage, 1 = Décoder si possible
cheminDocsJoint	Texte	→	Chemin du dossier (par défaut dans le dossier DocsJoint)
listePiècesJointes	Tab Alpha   Txt	←	Noms des fichiers joints (sans chemins d'accès)
Résultat	Entier	←	Code d'erreur

### Description

La commande MSG\_Extract extrait tous les documents joints et les place dans le dossier des documents joints.

nomFichier désigne le nom ou le chemin d'accès complet du fichier duquel extraire les documents joints. Si vous passez un nom de fichier seul, le chemin d'accès par défaut sera celui du dossier défini par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le chemin par défaut sera celui du dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou du dossier de 4D Client (avec 4D Server).

Le paramètre decoder spécifie si une tentative de décodage du ou des documents joints doit être effectuée. Si vous passez 0 (zéro), aucune tentative ne sera effectuée. Si vous passez 1, la commande décodera les fichiers, s'ils ont été encodés au moyen de l'une des méthodes suivantes : Binhex, AppleSingle, AppleDouble ou Base64.

Le paramètre cheminDocsJoint indique le chemin d'accès du dossier dans lequel enregistrer le document joint. Si vous passez une chaîne vide, le fichier est enregistré dans le dossier des documents joints spécifié par POP3\_SetPrefs. En l'absence de dossier spécifié par POP3\_SetPrefs, le fichier est enregistré dans le même dossier que celui du fichier de structure de la base de données.

Le tableau alpha/texte listePiècesJointes retourne les noms de tous les documents joints. Seul le nom du document est renvoyé dans l'élément du tableau, sans le chemin d'accès. Si vous utilisez un tableau alphanumérique, il doit être dimensionné pour contenir au moins 31 caractères, puisque la longueur maximale d'un nom de document est de 31 caractères.

### Référence

POP3\_SetPrefs.



MSG\_Delete (nomFichier{; dossier}) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Nom ou chemin d'accès de fichier
dossier	Entier	→	0 = Dossier Messages, 1 = Dossier DocsJoins
Résultat	Entier	←	Code d'erreur

### Description

La commande MSG\_Delete supprime un fichier local.

Le paramètre nomFichier indique le nom ou le chemin d'accès complet du fichier à supprimer.

Le paramètre dossier est utilisé lorsque nomFichier contient un nom de fichier seul. Il permet de désigner le dossier dans lequel la suppression doit être effectuée :

- 0 : dossier des messages (défini par POP3\_SetPrefs),
- 1 : dossier des documents joints (défini par POP3\_SetPrefs),

Dans les deux cas, en l'absence de dossier défini par POP3\_SetPrefs, le dossier utilisé sera celui contenant le fichier de structure de la base (avec 4D monoposte) ou le dossier de l'application 4D Client (avec 4D Server).

Si vous ne passez ni 0 ni 1 dans le paramètre dossier, par défaut la valeur 0 est utilisée.

**Attention** : Cette commande est à utiliser avec précaution, elle supprime **TOUT** fichier qui lui est transmis.

### Référence

POP3\_SetPrefs.



# 5

---

## IT File Transfer



Le protocole FTP (File Transfer Protocol) est le principal moyen de transférer documents et applications d'un ordinateur à un autre. Les "sites" FTP sont des ordinateurs dispersés dans le monde qui exécutent un logiciel FTP serveur. Le protocole FTP permet d'échanger des fichiers entre des systèmes disparates. Des applications clientes sur différentes plates-formes peuvent se connecter à un serveur FTP pour télécharger ou envoyer des fichiers texte ou binaires.

Les commandes FTP de 4D fournissent aux développeurs des outils leur permettant de créer des clients FTP à l'intérieur de leurs bases de données 4D.

**Note :** Lors de la spécification des chemins d'accès dans les commandes FTP, vous devez toujours définir les emplacements de fichiers sur les sites FTP comme étant des répertoires de type Unix, même si le serveur FTP est un Macintosh. Quelle que soit la plate-forme utilisée, le logiciel du serveur FTP convertit en interne ce chemin d'accès au format requis pour envoyer ses documents aux clients connectés.

FTP\_Progress (gauche; haut; titreFenêtre; texteThermo; annuler) → Entier

Paramètre	Type		Description
gauche	Entier	→	Coordonnée gauche de la fenêtre
haut	Entier	→	Coordonnée supérieure de la fenêtre
titreFenêtre	Alpha	→	Titre de la fenêtre du thermomètre
texteThermo	Alpha	→	Texte au-dessus du thermomètre
annuler	Alpha	→	Libellé du bouton Annuler
Résultat	Entier	←	Code d'erreur

**Description**

La commande FTP\_Progress définit les coordonnées de la fenêtre et le texte de la zone de dialogue de l'indicateur de progression FTP (thermomètre). Cet indicateur de progression s'affiche lors des appels à FTP\_Send, FTP\_Append ou à FTP\_Receive. La commande FTP\_Progress n'affiche pas la fenêtre de progression elle-même, elle définit seulement ses caractéristiques d'affichage. Les commandes FTP\_Send, FTP\_Append et FTP\_Receive permettent d'afficher ou de masquer cette fenêtre.

La fenêtre de progression se ferme automatiquement à la fin du transfert de fichier. Si pour une raison quelconque la taille du fichier en cours d'envoi ou de réception ne peut être déterminée, le thermomètre est affiché strié et la taille du fichier indique "unknown" (*inconnue*).

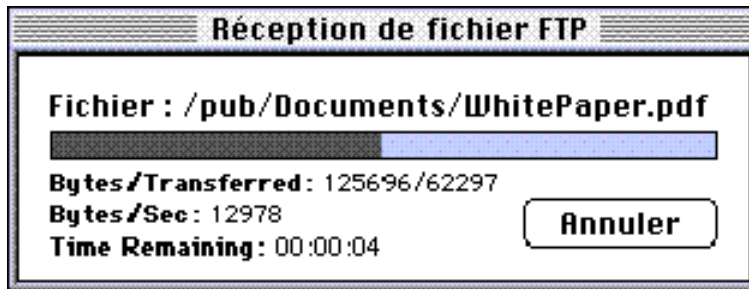
Le paramètre gauche indique la position du côté gauche de la fenêtre du thermomètre. Si vous passez -1 dans ce paramètre, la fenêtre est centrée horizontalement à l'écran.

Le paramètre haut indique la position du côté supérieur de la fenêtre du thermomètre de progression. Si vous passez -1 dans ce paramètre, la fenêtre est centrée verticalement à l'écran.

Le paramètre titreFenêtre vous permet de définir le titre de la fenêtre. Dans l'exemple suivant, le titre de la fenêtre est "Réception de fichier FTP". Si titreFenêtre est une chaîne vide, la fenêtre est créée sans titre.

Le paramètre `texteThermo` contient le texte à afficher au-dessus du thermomètre de progression. Dans l'exemple ci-après, `texteThermo` a pour valeur "Fichier : /pub/Documents/WhitePaper.pdf". Si vous passez "\*" dans ce paramètre, le texte d'état du transfert, envoyé par le serveur, est affiché (valeur par défaut). Ce texte varie en fonction des différentes étapes du processus de transfert.

Le paramètre `annuler` permet de définir le libellé du bouton **Annuler**. Si vous passez une chaîne vide, le bouton **Annuler** est masqué. Si vous passez "\*", le texte est celui par défaut, c'est-à-dire "Cancel".



### Exemple

```
⇒ $erreur:=FTP_Progress (-1;-1;"Réception de fichier FTP";"*","*")
   Au cas ou
       : (FTP_Login ("ftp.4d.com";"anonymous";"martin@societe.fr";vFTP_ID;
                    vFTP_TxtAccueil)#0)
       : (FTP_Receive(vFTP_ID;"/pub/Documents/WhitePaper.pdf";
                    "MonDisque:Docs:4D";1)#0)
       : (FTP_Logout (vFTP_ID)#0)
   Sinon
       $OK:=Vrai ` Toutes les commandes ont été exécutées sans erreur
   Fin de cas
```

FTP\_Login (nomServeur; nomUtilisateur; motDePasse; ftp\_ID{; texteAccueil}) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur
nomUtilisateur	Alpha	→	Nom de l'utilisateur
motDePasse	Alpha	→	Mot de passe
ftp_ID	Entier long	←	Référence de cette nouvelle session FTP
texteAccueil	Texte	←	Texte d'accueil FTP
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_Login établit une connexion avec le serveur FTP nomServeur et se connecte au système au moyen des nomUtilisateur et motDePasse fournis.

Le paramètre nomServeur contient le nom ou l'adresse IP du serveur distant.

nomUtilisateur contient le nom du compte utilisateur reconnu par le serveur FTP distant. De nombreux serveurs FTP acceptent l'accès d'invités au moyen du nom d'utilisateur "anonymous". Si vous vous connectez de façon anonyme, il est de règle de fournir votre adresse e-mail comme mot de passe.

Le paramètre motDePasse contient le mot de passe de nomUtilisateur sur le serveur FTP.

Le paramètre ftp\_ID retourne un entier long identifiant la session ouverte. Cette valeur sera utilisée par les commandes FTP suivantes.

Le paramètre optionnel texteAccueil récupère le texte envoyé par le serveur FTP lorsque l'utilisateur se connecte. En effet, de nombreux sites FTP envoient un message d'accueil lors de la connexion.



## Exemple

```
$OK:=Faux
Au cas ou
⇒      : (FTP_Login ("ftp.4d.com";"anonymous";"dbody@aol.com";vFTP_ID;
          vFTP_TxtAccueil)#0)
        : (FTP_Progress (-1;-1;"Fenêtre de progression";"Réception du fichier
          demandé...";"*)#0)
        : (FTP_Send (vFTP_ID;"Mon disque dur:Documents f:Rapport des ventes de juillet";
          "/pub/rapports";1)#0)
        : (FTP_Logout (vFTP_ID)#0)
Sinon
      $OK:=Vrai  ` Toutes les commandes ont été exécutées sans erreur
Fin de cas
```

## Référence

FTP\_Logout.

FTP\_GetDirList (ftp\_ID; cheminServeur; noms; tailles; types; datesModif) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès à un répertoire Unix sur le serveur FTP
		←	Répertoire de travail courant (CWD)
noms	Tab Alpha   Txt	←	Liste de noms
tailles	Tab Entier long	←	Liste de tailles
types	Tab Entier	←	Liste de types 0 = fichier normal, 1 = répertoire, 2 = fichier spécial de type bloc, 3 = fichier spécial de type caractère, 4 = lien symbolique, 5 = fichier spécial FIFO, 6 = porte d'accès de la famille d'adresses AF_UNIX
datesModif	Tab Date	←	Liste des dates de modification
Résultat	Entier	←	Code d'erreur

Description

La commande FTP\_GetDirList retourne la liste des objets présents dans le répertoire cheminServeur de la session FTP identifiée par ftp\_ID. Le nom, la taille, le type et la date de modification des éléments du répertoire cheminServeur sont renvoyés dans quatre tableaux. Une connexion au site FTP doit avoir déjà été ouverte par FTP\_Login et être toujours valide (FTP\_VerifyID). La commande FTP\_GetDirList remplace le répertoire de travail courant (ou CWD, *Current Working Directory*) par celui défini dans le paramètre cheminServeur.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre `cheminServeur` contient un chemin d'accès au format Unix référençant un répertoire FTP. Il est fortement recommandé de passer une variable ou un champ 4e Dimension dans ce paramètre car le répertoire de travail courant résultant lui sera retourné après l'exécution de la commande. Généralement, la valeur renvoyée sera identique à celle qui a été passée. Toutefois, dans certains cas (par exemple des restrictions d'accès), le changement de répertoire a échoué. Le paramètre `cheminServeur` contient alors le chemin d'accès du répertoire courant du serveur pour la session. Si vous passez une chaîne vide dans ce paramètre, les tableaux sont remplis avec la liste des fichiers du répertoire courant et le chemin d'accès du répertoire courant du serveur (CWD) est retourné dans le paramètre `cheminServeur`.

`noms` est un tableau de type alphanumérique ou texte recevant le nom de chaque objet présent dans le répertoire `cheminServeur` spécifié.

`tailles` est un tableau de type entier long recevant la taille des objets du répertoire `cheminServeur`.

`types` est un tableau de type entier recevant les valeurs de type de chaque objet du répertoire `cheminServeur`. Voici les valeurs possibles et les types correspondants :

Type	Fichier
0	fichier ordinaire
1	répertoire
2	fichier spécial de type bloc
3	fichier spécial de type caractère
4	lien symbolique
5	fichier spécial FIFO
6	porte d'accès de la famille AF_UNIX

`datesModif` est un tableau de type dates recevant la date de dernière modification de chaque objet du répertoire `cheminServeur`.

## Référence

FTP\_ChangeDir, FTP\_Login, FTP\_PrintDir, FTP\_VerifyID.

FTP\_ChangeDir (ftp\_ID; cheminServeur) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès à un répertoire Unix sur le serveur FTP
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_ChangeDir permet de désigner le répertoire de travail courant (*Current Working Directory* ou CWD) sur le serveur FTP.

**Note :** Il est également possible de modifier le répertoire de travail courant à l'aide des commandes FTP\_GetDirList et FTP\_GetFileInfo. Toutefois, la commande FTP\_ChangeDir est plus rapide et nécessite moins de paramètres.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre cheminServeur contient un chemin d'accès au format Unix référençant un répertoire FTP existant et accessible. Si le répertoire FTP spécifié est invalide (inexistant ou droits d'accès insuffisants), FTP\_ChangeDir retourne une erreur et ne modifie pas le répertoire de travail courant.

### Exemple

L'exemple suivant désigne la racine du serveur FTP comme répertoire de travail courant:

⇒ `$err=FTP_ChangeDir($ftp_ID;"/")`

### Référence

FTP\_GetDirList, FTP\_GetFileInfo, FTP\_PrintDir.

FTP\_PrintDir (ftp\_ID; cheminServeur) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	←	Chemin d'accès Unix à un répertoire du serveur FTP
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_PrintDir retourne le répertoire de travail courant (*Current Working Directory*, ou CWD) sur le serveur FTP.

**Note** : Cette information peut également être obtenue à l'aide de l'instruction FTP\_GetDirList (ftpID;"";...) en passant une chaîne vide dans le paramètre cheminServeur. Toutefois, la commande FTP\_PrintDir est plus rapide et nécessite moins de paramètres.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre cheminServeur retourne le chemin du répertoire de travail courant (CWD).

### Exemple

L'emplacement du CWD est retourné dans la variable \$Cwd:

⇒ `$err:=FTP_PrintDir($myftpid;$Cwd)`

### Référence

FTP\_GetDirList.

FTP\_GetFileInfo (ftp\_ID; cheminServeur; taille; dateModif) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès au document sur le serveur FTP
taille	Entier long	←	Taille du document
dateModif	Date	←	Date de modification
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_GetFileInfo retourne la taille et la date de dernière modification du fichier désigné par cheminServeur.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

cheminServeur contient le chemin d'accès au document sur lequel vous souhaitez obtenir des informations.

**Note :** La commande FTP\_GetFileInfo peut modifier le répertoire de travail courant (CWD) si cheminServeur est un chemin d'accès complet qui indique un répertoire différent du répertoire de travail courant. Dans ce cas, le répertoire défini par le paramètre cheminServeur devient le répertoire de travail courant.

Le paramètre taille retourne la taille du fichier identifié par cheminServeur.

Le paramètre dateModif retourne la date de la dernière modification du fichier.

### Référence

FTP\_GetDirList.

FTP\_VerifyID (ftp\_ID) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→ ←	Référence d'une connexion FTP 0 = La connexion est déjà fermée
Résultat	Entier	←	Code d'erreur

### Description

Un serveur FTP déconnecte automatiquement les comptes demeurés inactifs pendant une certaine période, déterminée par son administrateur (ce mécanisme est appelé *timeout*).

Chaque commande en interaction avec le serveur FTP provoque la remise à zéro du compteur d'inactivité.

La commande FTP\_VerifyID provoque également la remise à zéro du compteur d'inactivité de la session FTP spécifiée. Elle permet de conserver une session active en cas de risque de dépassement du délai.

Lors de son exécution, la commande FTP\_VerifyID vérifie que la connexion n'a pas déjà été close. Si la session est toujours ouverte, la commande indique au serveur FTP de remettre à zéro le minuteur d'inactivité pour la session. Si la connexion a déjà été fermée, FTP\_VerifyID retourne l'erreur appropriée, libère la mémoire utilisée par la session FTP et renvoie 0 dans le paramètre ftp\_ID.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

### Référence

FTP\_Login.

FTP\_MakeDir (ftp\_ID; cheminServeur) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès à un répertoire Unix sur le serveur FTP
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_MakeDir crée un nouveau répertoire défini par le paramètre cheminServeur. Une erreur est retournée si vous n'avez pas les droits d'accès requis pour effectuer cette opération.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

cheminServeur désigne le chemin d'accès au format Unix du répertoire FTP à créer. Ce paramètre peut contenir un chemin d'accès complet ou un simple nom de dossier. Si la forme abrégée est utilisée, le répertoire est alors créé dans le répertoire de travail courant (*CWD*). Le nom du répertoire cheminServeur ne doit **pas** comporter d'espace vide.

**Note :** La commande FTP\_ChangeDir permet de modifier le répertoire de travail courant (*CWD*). Vous pouvez également connaître à tout moment le répertoire de travail courant à l'aide de la commande FTP\_PrintDir.

### Référence

FTP\_ChangeDir, FTP\_GetDirList, FTP\_PrintDir, FTP\_RemoveDir.



FTP\_RemoveDir (ftp\_ID; cheminServeur) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès à un répertoire Unix sur le serveur FTP
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_RemoveDir supprime le répertoire désigné par le paramètre cheminServeur. Une erreur est retournée si vous n'avez pas les droits d'accès requis pour effectuer cette opération. En outre, si vous tentez de supprimer un répertoire contenant des éléments, une erreur de sécurité sera également retournée.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

cheminServeur désigne le chemin d'accès au format Unix d'un répertoire FTP à supprimer. Ce paramètre peut contenir un chemin d'accès complet ou un simple nom de dossier. Si la forme abrégée est utilisée, le répertoire spécifié doit se trouver dans le répertoire de travail courant (*CWD*).

**Note :** La commande FTP\_ChangeDir permet de modifier le répertoire de travail courant (*CWD*). Vous pouvez également connaître à tout moment le répertoire de travail courant à l'aide de la commande FTP\_PrintDir.

### Référence

FTP\_GetDirList, FTP\_MakeDir.

FTP\_Rename (ftp\_ID; cheminServeur; nouvNom) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès du document sur le serveur FTP
nouvNom	Texte	→	Nouveau nom du document
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_Rename renomme le fichier désigné par cheminServeur sur le serveur FTP distant. Une erreur est générée si vous n'avez pas les droits d'accès requis pour effectuer cette opération.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

cheminServeur désigne le chemin d'accès au format Unix du document à renommer. Vous pouvez passer dans ce paramètre un chemin d'accès complet ou un simple nom de fichier. Si vous utilisez la forme abrégée, le fichier spécifié doit se trouver dans le répertoire de travail courant (*CWD*).

**Note :** La commande FTP\_ChangeDir permet de modifier le répertoire de travail courant (*CWD*). Vous pouvez également connaître à tout moment le répertoire de travail courant à l'aide de la commande FTP\_PrintDir.

Le paramètre nouvNom contient le nom avec lequel vous souhaitez renommer le document distant.

### Référence

FTP\_ChangeDir, FTP\_PrintDir.

FTP\_Delete (ftp\_ID; cheminServeur) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès du document sur le serveur FTP
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_Delete supprime le fichier désigné par cheminServeur du serveur FTP distant. Une erreur est renvoyée si vous n'avez pas les droits d'accès requis pour effectuer cette opération.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

cheminServeur désigne le chemin d'accès au format Unix du document à supprimer. Vous pouvez passer dans ce paramètre un chemin d'accès complet ou un simple nom de fichier. Si vous utilisez la forme abrégée, le fichier spécifié doit se trouver dans le répertoire de travail courant (*CWD*).

**Note :** La commande FTP\_ChangeDir permet de modifier le répertoire de travail courant (*CWD*). Vous pouvez également connaître à tout moment le répertoire de travail courant à l'aide de la commande FTP\_PrintDir.

### Référence

FTP\_ChangeDir, FTP\_PrintDir, FTP\_RemoveDir.

FTP\_MacBinary (ftp\_ID; modeMacBinary) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
modeMacBinary	Entier	→	-1 = Obtenir le paramètre courant, 1 = Activer, 0 = Désactiver
		←	Paramètre courant (si -1 passé)
Résultat	Entier	←	Code d'erreur

Description

La commande FTP\_MacBinary active/désactive le mode MacBinary lors des transferts FTP utilisant FTP\_Send ou FTP\_Receive dans la session FTP courante identifiée par ftp\_ID.

Le protocole MacBinary est souvent utilisé par les clients et serveurs FTP Macintosh pour faciliter le transfert de données ou de fichiers binaires contenant à la fois des *data forks* (parties de "données") et des *resource forks* (parties de "ressources").

**Note à l'attention des utilisateurs Windows :** Il est possible d'utiliser le protocole MacBinary pour des transferts FTP dans un environnement Windows. Toutefois, il est généralement inutile de décoder un fichier MacBinary sur un PC. Les ordinateurs à base de processeur Intel ne peuvent pas stocker des fichiers contenant une *data forks* et une *resource fork* (format MacOS). Comme ce format est étranger à la plate-forme PC, les fichiers MacOS contenant une *resource fork* risquent d'être endommagés s'ils sont enregistrés dans un format non encodé.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre modeMacBinary indique s'il faut activer ou non le mode de transfert MacBinary. Cette valeur doit être fournie en tant que variable pour que la commande puisse renvoyer l'état des transferts MacBinary après une tentative de modification.

- 1 active le mode de transfert MacBinary.
- 0 (zéro) désactive le mode de transfert MacBinary.
- -1 retourne dans modeMacBinary le paramétrage courant du mode de transfert MacBinary (1 ou 0).

**Attention :** Tous les serveurs FTP ne gèrent pas le protocole MacBinary. S'il n'est pas géré, l'erreur 10053 est générée à chaque appel de la commande FTP\_MacBinary, quelle que soit la valeur du paramètre modeMacBinary. Les comportements précédemment décrits ne s'appliquent alors pas.

## Exemple

Cet exemple active le protocole MacBinary avant la réception d'un fichier FTP. Si le fichier a été correctement reçu avec MacBinary activé, il est alors décodé dans son format original et le document MacBinary reçu est supprimé.

```
vUtiliserMacBin:=-1
⇒ $erreur:=FTP_MacBinary (vFTP_ID;vUtiliserMacBin)
  Si($erreur=10053)
    MacBinaryEstAccepte:=Faux `Le serveur ftp ne gère pas le protocole MacBinary
  Sinon
    MacBinaryEstAccepte:=Vrai
  Fin de si

vFichierLocal:=""
Si(MacBinaryEstAccepte)
  vUtiliserMacBin:=1
  `Activation de MacBinary pour le téléchargement
⇒ $erreur:=FTP_MacBinary (vFTP_ID;vUtiliserMacBin)
  Fin de si
  $erreur:=FTP_Receive (vFTP_ID;"MonApplication";vFichierLocal;cbShowTherm)
  `Si la réception est OK et le fichier est au format MacBinary
  Si ($erreur=0) & (vUtiliserMacBin=1)
    vCheminDecode:=""
    Si (IT_Decode (vFichierLocal;vCheminDecode;8)=0) `Décodage MacBinary
      `Si le décodage a réussi, alors supprimer le fichier source
      SUPPRIMER DOCUMENT(vFichierLocal)
    Fin de si
  Fin de si
```

## Référence

IT\_Decode.

FTP\_Send (ftp\_ID; cheminLocal; cheminServeur; progression) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminLocal	Texte	→	Chemin d'accès local du document à envoyer
cheminServeur	Texte	→	Chemin d'accès du document sur le serveur FTP
progression	Entier	→	1 = Montrer progression, 0 = Masquer progression
Résultat	Entier	←	Code d'erreur

**Description**

La commande FTP\_Send envoie le document désigné par cheminLocal à l'emplacement désigné par cheminServeur. Si une erreur de statut de fichier FTP se produit, FTP\_Send la retourne immédiatement.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

cheminLocal désigne le chemin d'accès local du document à envoyer. Si vous passez une chaîne vide, la boîte de dialogue standard d'ouverture de fichiers apparaît. Si vous passez un nom de fichier simple (sans chemin d'accès), la commande recherche ce fichier dans le dossier contenant le fichier de structure de la base de données (avec 4D monoposte) ou dans le dossier de 4D Client (avec 4D Server). Comme pour tous les chemins d'accès à des documents locaux, les noms des répertoires doivent être séparés par le délimiteur correspondant à la plate-forme. Pour plus d'informations sur ce point, reportez-vous à la section Glossaire et terminologie.

cheminServeur désigne le chemin d'accès complet au format Unix du document à créer, nom du fichier compris. Le cheminServeur indique le nouveau nom du fichier une fois reçu par le serveur FTP. Si cheminLocal est une chaîne vide permettant à l'utilisateur de sélectionner un fichier sur disque, cheminServeur peut alors aussi être une chaîne vide, auquel cas le nom du fichier sélectionné sera utilisé.

Vous pouvez passer dans `cheminServeur` un chemin d'accès Unix complet ou simplement un nom de fichier :

- Si vous passez un chemin d'accès complet, le fichier sera placé dans le répertoire défini, qui deviendra alors le répertoire de travail courant (CWD).
- Si vous passez uniquement un nom de fichier, ou si des chaînes vides sont passées dans les paramètres `cheminLocal` et `cheminServeur`, le fichier sera alors envoyé dans le répertoire de travail courant (CWD).

Si le paramètre `cheminServeur` ne peut pas être correctement interprété, ou si l'utilisateur n'a pas les droits d'accès requis pour envoyer un fichier dans le répertoire distant, la commande retourne une erreur. Comme pour tous les chemins d'accès des documents Unix, le chemin doit être défini à l'aide de barres obliques ("/"). Pour plus d'informations sur ce point, reportez-vous à la section Glossaire et terminologie.

Le paramètre `progression` vous permet de spécifier si le thermomètre de progression doit ou non être affiché. Passez 1 pour l'afficher et 0 (zéro) pour le masquer.

### Exemples

- Exemple 1

```
$OK:=Faux
Au cas ou
: (FTP_Login ("ftp.4d.com";"anonymous";dbody@aol.com;vFTP_ID;vAccueil)#0)
: (FTP_Progress (-1;-1;"Fenêtre de progression";"Réception du fichier demandé...";
"Annuler")#0)
⇒ : (FTP_Send (vFTP_ID;"Mon disque dur:Documents:Rapport des ventes de juillet";
"/pub/rapports/";1)#0)
: (FTP_Logout (vFTP_ID)#0)
Sinon
$OK:=Vrai `Toutes les commandes ont été exécutées sans erreur
Fin de cas
```

- Exemple 2

```
⇒ $erreur:=FTP_Send (vFTP_ID;"";"";1)
```

### Référence

FTP\_Progress, FTP\_Receive.

FTP\_Append (ftp\_ID; cheminLocal; cheminServeur; progression) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminLocal	Texte	→	Chemin d'accès local du document à envoyer
cheminServeur	Texte	→	Chemin d'accès du document sur le serveur FTP
progression	Entier	→	1 = Montrer progression, 0 = Masquer progression
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_Append effectue la même action que FTP\_Send, à la différence près qu'elle ajoute les données envoyées à la fin du fichier existant identifié par cheminServeur. La principale fonction de cette commande est d'ajouter des données à la fin de fichiers texte préexistants.

### Référence

FTP\_Send.



FTP\_GetType (ftp\_ID; ftp\_Mode) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
ftp_Mode	Alpha	←	"A" = Ascii ; "I" = Image ; "L 8" = 8 bits Logique
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_GetType renvoie des informations sur le mode de transfert FTP courant. Le mode de transfert peut être fixé au moyen de la commande FTP\_SetType.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre ftp\_Mode retourne un code indiquant le mode de transfert FTP courant.

### Référence

FTP\_SetType.

FTP\_SetType (ftp\_ID; ftp\_Mode) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
ftp_Mode	Alpha	→	"A" = Ascii ; "I" = [défaut] Image ; "L 8" = 8 bits Logique
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_SetType permet de modifier le mode de transfert FTP utilisé pendant les opérations d'envoi/réception. En général, il n'est pas nécessaire de modifier ce paramètre.

Toutefois, en raison des différences existant entre les diverses plates-formes et versions de FTP, le changement de mode de transfert peut, dans certains cas, s'avérer utile. Par exemple, le transfert de documents en texte seul peut nécessiter le mode Ascii.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Passez dans le paramètre ftp\_Mode le code du mode de transfert à utiliser. Ce paramétrage est pris en compte pour toutes les opérations d'envoi/réception ultérieures. Par défaut, le mode Image ("I") est utilisé.

### Référence

FTP\_GetType.

FTP\_System (ftp\_ID; infoSystème) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
infoSystème	Alpha	←	Informations sur le système
Résultat	Entier	←	Code d'erreur

**Description**

La commande FTP\_System retourne dans le paramètre infoSystème des informations décrivant le logiciel du serveur FTP.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

FTP\_Receive (ftp\_ID; cheminServeur; cheminLocal; progression) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
cheminServeur	Texte	→	Chemin d'accès sur le serveur FTP du document à recevoir
cheminLocal	Texte	→	Chemin d'accès local de destination du document
		←	Chemin d'accès du document résultant (si "" passé)
progression	Entier	→	0 = Masquer progression, 1 = Afficher progression
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_Receive reçoit par FTP un fichier dont le chemin d'accès sur le serveur FTP est défini par cheminServeur. FTP\_Receive retourne l'erreur -48 si le fichier existe déjà dans le répertoire de destination.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre cheminServeur spécifie le chemin d'accès Unix complet du document à recevoir. Si cheminServeur ne contient pas un chemin d'accès complet à un document, la commande retourne une erreur. Comme pour tous les chemins d'accès à des documents Unix, le chemin doit utiliser des barres obliques ("/") comme séparateurs. Pour plus d'informations sur ce point, reportez-vous à la section Glossaire et terminologie.

Le paramètre cheminLocal spécifie le chemin d'accès du document à créer localement.

- Si vous passez une chaîne vide dans ce paramètre, la boîte de dialogue standard d'enregistrement de document s'affichera — dans ce cas, le nom et le chemin d'accès du fichier sélectionné par l'utilisateur seront retournés dans la variable cheminLocal.

- Si vous passez uniquement un nom de fichier, le fichier sera enregistré dans le même dossier que le fichier de structure de la base de données (avec 4D monoposte) ou dans le dossier de 4D Client (avec 4D Server).

Comme pour tous les chemins d'accès à des documents locaux, les répertoires doivent être séparés par le délimiteur correspondant à la plate-forme utilisée. Pour plus d'informations, reportez-vous à la section Glossaire et terminologie.

Le paramètre progression vous permet d'indiquer si le thermomètre de progression doit ou non être affiché durant la réception. Passez 1 pour afficher le thermomètre, ou 0 pour le masquer.

## Exemple

```
vUtiliserMacBin:=-1
$erreur:=FTP_MacBinary (vFTP_ID;vUtiliserMacBin)
Si($erreur=10053)
    MacBinaryEstAccepte:=Faux    `Le serveur ftp ne gère pas le protocole MacBinary
Sinon
    MacBinaryEstAccepte:=Vrai
Fin de si

vFichierLocal:=""
Si(MacBinaryEstAccepte)
    vUtiliserMacBin:=1
    `Active MacBinary pour le téléchargement
    $erreur:=FTP_MacBinary (vFTP_ID;vUtiliserMacBin)
⇒ $erreur:=FTP_Receive (vFTP_ID;"CGMiniViewer.hqx";vFichierLocal;cbShowTherm)
    Si ($erreur=0) & (vUtiliserMacBin=1)
        vCheminDecode:=""
        Si (IT_Decode (vFichierLocal;vCheminDecode;8)=0) `Décoder MacBinary
            `Si le décodage a réussi, supprimer le fichier source
            SUPPRIMER DOCUMENT(vFichierLocal)
        Fin de si
    Fin de si
Fin de si
```

## Référence

FTP\_MacBinary, IT\_Decode.

FTP\_Logout (ftp\_ID) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
		←	0 = Session correctement fermée
Résultat	Entier	←	Code d'erreur

### Description

La commande FTP\_Logout déconnecte l'utilisateur du serveur FTP et libère la mémoire utilisée pour la session FTP désignée par ftp\_ID. Après son exécution, la commande retourne la valeur 0 (zéro) dans le paramètre ftp\_ID si la fermeture de la session a réussi.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

### Exemple

```
Si (FTP_Login ("ftp.4d.com";"anonymous";"dboy@aol.com";vFTP_ID;Txt_Accueil)=1)
    $erreur:=FTP_Send (vFTP_ID;"Mon disque dur:Documents:Rapports des ventes";
"/pub/rapports";1)
⇒      $erreur:=FTP_Logout (vFTP_ID)
Fin de si
```

### Référence

FTP\_Login.

---

FTP\_SetPassive (ftp\_ID; modePassif) → Entier

Paramètre	Type	Description
ftp_ID	Entier long →	Référence d'une connexion FTP
modePassif	Entier →	0 = mode passif (mode par défaut), 1 = mode actif

### Description

La commande FTP\_SetPassive permet de définir les modalités d'échange sur le canal de transfert des données entre un serveur et un client FTP, lors de l'utilisation des commandes telles que FTP\_GetDirList, FTP\_Send, FTP\_Append et FTP\_Receive. Le mode de transfert spécifié est utilisé par ces commandes dès que vous avez exécuté FTP\_SetPassive.

Les échanges entre un serveur et un client FTP se décomposent en deux parties : échanges sur le canal de contrôle (port 21 par défaut) et échanges sur le canal de transfert de données (port 20 par défaut).

Généralement, les serveurs FTP sont dits "actifs", car ils prennent en charge la gestion du transfert sur le canal de données et se connectent au client FTP.

Pour des raisons historiques, les commandes Internet de 4D demandent aux serveurs FTP de fonctionner en mode passif, ce qui consiste notamment à exécuter systématiquement la commande FTP "PASV" avant chaque échange sur le canal de transfert de données.

Toutefois, certains serveurs FTP ne permettent pas l'emploi du mode passif. En outre, ce mode peut être interdit par les systèmes de protection (*firewall*) des sites. Dans ce cas, la commande FTP\_SetPassive vous permet de définir le mode actif pour le transfert des données.

**Note :** Il est recommandé de vérifier avec l'administrateur réseau le mode de transfert préconisé sur le site.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre modePassif permet de préciser le mode de transfert sur le canal de transfert des données :

- Si vous passez 0, vous demandez au serveur FTP de fonctionner en mode actif.
- Si vous passez 1, vous demandez au serveur FTP de fonctionner en mode passif (valeur par défaut).

### Référence

FTP\_GetPassive.

FTP\_GetPassive (ftp\_ID; modePassif) → Entier

Paramètre	Type		Description
ftp_ID	Entier long	→	Référence d'une connexion FTP
modePassif	Entier	←	Mode d'échange en cours 0 = mode actif, 1 = mode passif

### Description

La commande FTP\_GetPassive permet de connaître le mode de transfert en vigueur sur le canal de transfert des données. Pour plus d'informations sur ce paramétrage, reportez-vous à la description de la commande FTP\_SetPassive.

ftp\_ID est l'identifiant de la session FTP établie avec FTP\_Login.

Le paramètre modePassif retourne le mode de transfert courant sur le canal de transfert des données :

- si modePassif vaut 0, le serveur FTP fonctionne en mode actif.
- si modePassif vaut 1, le serveur FTP fonctionne en mode passif (mode par défaut).

### Référence

FTP\_SetPassive.



# 6

---

## IT TCP/IP



TCP/IP ou Transmission Control Protocol/Internet Protocol est le principal protocole utilisé pour l'envoi de données sur Internet. Les commandes Internet de bas niveau de 4D permettent d'établir des sessions TCP puis d'envoyer et/ou de recevoir des paquets TCP via ces sessions.

Il existe deux manières d'établir une connexion TCP :

- La première consiste à exécuter la commande `TCP_Open`. Celle-ci ouvre une connexion avec le domaine et sur le port spécifiés. Elle permet de se connecter à un serveur TCP.
- La seconde consiste à exécuter la commande `TCP_Listen`. Celle-ci ouvre une connexion avec le domaine et sur le port spécifiés, et écoute les connexions entrantes. La meilleure façon de déterminer si une connexion a été établie consiste à vérifier l'état de la session avec la commande `TCP_State`, après l'exécution de `TCP_Listen`. Un code de statut est alors renvoyé, indiquant l'état courant de la session. A partir de là, vous pouvez envoyer et/ou recevoir des paquets TCP comme vous le feriez lors d'une connexion établie avec `TCP_Open`.

Les commandes TCP/IP de bas niveau nécessitent une connaissance approfondie des protocoles de communication. Pour obtenir des informations complémentaires sur les différents numéros de port affectés à TCP/IP, les protocoles de communication, les impératifs d'adressage, etc., veuillez vous reporter aux RFCs.

TCP\_Open (nomServeur; portDistant; tcp\_ID) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur
portDistant	Entier	→	Port distant auquel se connecter (0 = indifférent)
tcp_ID	Entier long	←	Référence de la session TCP ouverte
Résultat	Entier	←	Code d'erreur

### Description

La commande TCP\_Open ouvre une connexion TCP sortante vers un domaine.

TCP\_Open établit une connexion avec nomServeur, sur le port référencé par portDistant (si ce paramètre est différent de 0), et retourne le numéro de la session dans tcp\_ID. Cette valeur sera utilisée par tous les appels TCP ultérieurs se rapportant à cette session. Par défaut, une session ouverte par TCP\_Open n'est maintenue que pendant 30 secondes (*timeout*) si aucune donnée n'est reçue. Cette valeur de *timeout* par défaut peut être modifiée à l'aide de la commande IT\_SetTimeout.

Le paramètre nomServeur contient le nom ou l'adresse IP du serveur avec lequel vous ouvrez une connexion.

Le paramètre portDistant permet d'indiquer le port TCP de la machine désignée par nomServeur, avec laquelle vous souhaitez établir une connexion. Pour utiliser n'importe quel port, passez 0 (zéro) dans ce paramètre.

**Note :** Après un appel à TCP\_Open (ou TCP\_Listen), portDistant peut retourner une valeur négative si la valeur passée dans ce paramètre est supérieure à 32767. Cela ne perturbe pas la connexion. Toutefois, pour éviter cette situation, vous pouvez utiliser une variable intermédiaire :

```
$v_ portDistant:=v_ portDistant
```

```
$err:=TCP_Open (v_ AdrIPServeurDistant;$v_ portDistant;v_ SessionID)
```

tcp\_ID retourne la référence de la session ouverte. Cette référence sera utilisée par toutes les commandes TCP ultérieures exécutées dans la session.

### Référence

IT\_SetTimeout.

---

TCP\_Listen (clientDistant; portLocal; portDistant; timeOut; tcp\_ID) → Entier

Paramètre	Type		Description
clientDistant	Alpha	→	Nom ou adresse IP du client
portLocal	Entier	→	Numéro de port local, 0 = Utiliser un port local vacant
		←	Numéro du port local utilisé (si 0 passé)
portDistant	Entier	→	Numéro du port à écouter
timeOut	Entier	→	Nombre de secondes à attendre avant timeout, 0 = Pas de timeout
tcp_ID	Entier long	←	Référence de la session TCP ouverte
Résultat	Entier	←	Code d'erreur

### Description

La commande TCP\_Listen attend une connexion de la machine désignée par clientDistant depuis le port portDistant.

Cette commande ne rend pas la main à la méthode d'appel de 4e Dimension tant que la connexion n'a pas été ouverte ou que la période de timeOut ne s'est pas écoulée. L'exécution de TCP\_Listen peut donc provoquer temporairement le blocage de la base. Toutefois, cette commande est particulièrement adaptée au multi-process : elle partage le temps CPU avec les autres process 4D en cours d'exécution. Il est donc conseillé d'exécuter TCP\_Listen au sein d'un process 4D particulier (en particulier si aucun timeOut n'est spécifié).

Le paramètre clientDistant contient le nom ou l'adresse IP du serveur duquel vous attendez une connexion. Si vous passez une chaîne vide dans ce paramètre, la commande accepte toute connexion entrante en provenance de toute machine.

Le paramètre portLocal contient le numéro de port de la machine locale à utiliser pour la communication. Si vous passez 0 (zéro), la commande utilisera un port vacant et retournera son numéro dans ce paramètre.

portDistant contient le numéro du port du serveur depuis lequel écouter la connexion.

**Note :** Après un appel à TCP\_Listen (ou TCP\_Open), portDistant peut retourner une valeur négative si la valeur transmise à ce paramètre est supérieure à 32767. Cela ne perturbe pas la connexion. Toutefois, pour éviter cette situation, vous pouvez utiliser une variable intermédiaire :

```
$v_ portDistant:=v_ portDistant  
$err:=TCP_Listen (v_ AdrIPServeurDistant;0;$v_ portDistant;30;v_ SessionID)
```

Le paramètre timeOut spécifie le nombre maximum de secondes pendant lequel la commande attendra une connexion entrante. Si vous passez 0 (zéro), la commande attendra indéfiniment la connexion (pas de *timeout*). Utilisez cette option avec précaution car le contrôle n'est pas restitué au process d'appel 4D si aucune connexion n'est effectuée. En particulier, ne passez pas 0 dans ce paramètre si la base est mono-process.

Le paramètre tcp\_ID retourne la référence de la session ouverte. Cette référence sera utilisée par toutes les commandes TCP ultérieures exécutées dans la session.

### Exemple

```
      C_ENTIER LONG(vTCPID)  
      C_ENTIER(vStatus)  
⇒    $err:=TCP_Listen ("";0;49152;30;vTCPID)  
      $err:=TCP_State (vTCPID;vStatut)  
      Si (vStatut=8) `La connexion a été établie  
          FaireQuelqueChose  
          $err:=TCP_Close (vTCPID)  
      Fin de si
```

### Référence

Annexe B, Numéros des ports TCP, TCP\_Open, TCP\_State.

TCP\_Send (tcp\_ID; envoyerTexte) → Entier

Paramètre	Type		Description
tcp_ID	Entier long	→	Référence de session TCP ouverte
envoyerTexte	Texte	→	Texte à envoyer
Résultat	Entier	←	Code d'erreur

### Description

La commande TCP\_Send envoie des données à la session TCP désignée par tcp\_ID.

tcp\_ID est la référence d'une session TCP ouverte, établie avec la commande TCP\_Open ou TCP\_Listen.

Le paramètre envoyerTexte contient une valeur de type texte à envoyer à la session TCP référencée par tcp\_ID.

### Référence

TCP\_Listen, TCP\_Open, TCP\_SendBlob.

TCP\_Receive (tcp\_ID; texte) → Entier

Paramètre	Type		Description
tcp_ID	Entier long	→	Référence à une session TCP ouverte
texte	Texte	←	Texte reçu
Résultat	Entier	←	Code d'erreur

**Description**

La commande TCP\_Receive permet de recevoir des paquets de données au cours d'une session TCP.

Passez dans tcp\_ID la référence d'une session TCP ouverte avec la commande TCP\_Open ou TCP\_Listen.

Le paramètre texte retourne sous forme de texte les données reçues.

Lors de la réception de données par paquets TCP, vous ne pouvez pas avoir la certitude que la totalité des données sont reçues par un seul appel TCP\_Receive. La commande TCP\_Receive est donc généralement appelée dans une boucle 'Repeter' qui vérifie en permanence le statut de la connexion ou attend une valeur particulière.

**Exemple**

```
C_ENTIER LONG($tcp_id)
C_TEXTE($pageweb;$tampon)
C_ENTIER(vEtat;$erreur)
$pageweb:=""
vEtat:=0
Repeter
⇒ $erreur:=TCP_Receive ($tcp_id;$tampon)
   $erreur:=TCP_State ($tcp_id;vEtat)
   $pageweb:=$pageweb+$tampon
Jusque ((vEtat=0) | ($erreur#0)) `Jusqu'à ce que le serveur ferme la connexion,
                                ` ou qu'une erreur soit retournée
```

**Référence**

TCP\_Send, TCP\_SendBlob.



TCP\_SendBLOB (tcp\_ID; blobAEnvoyer) → Entier

Paramètre	Type		Description
tcp_ID	Entier long	→	Référence à une session TCP ouverte
blobAEnvoyer	BLOB	→	Données à envoyer
Résultat	Entier	←	Code d'erreur

### Description

La commande TCP\_SendBLOB envoie des données à la session TCP désignée par tcp\_ID. Elle fonctionne de manière semblable à la commande TCP\_Send, à la différence près qu'elle envoie des données de type BLOB au lieu d'un texte. Ce fonctionnement permet de s'affranchir de la limite des 32 000 caractères inhérente aux données de type Texte, et ainsi d'envoyer des données binaires.

tcp\_ID est la référence d'une session TCP ouverte, établie avec la commande TCP\_Open ou TCP\_Listen.

Le paramètre blobAEnvoyer contient les données de type BLOB à envoyer à la session TCP référencée par tcp\_ID.

**Note sur l'indépendance de plate-forme :** Lors de l'envoi de données binaires dans un format propriétaire, il vous appartient de traiter les conversions d'octets ("byte swapping") entre les différentes plates-formes, si nécessaire.

### Exemple

Cet exemple place du texte dans un BLOB puis l'envoie dans la session TCP :

```
C_BLOB($Blob_AEnvoyer)
C_TEXTE(v_Txt_Send)
TEXTE VERS BLOB(v_Txt_Send;$Blob_AEnvoyer;Texte sans longueur;)
⇒ $err:=TCP_SendBLOB (v_tcp_ID;$Blob_AEnvoyer)
```

### Référence

TCP\_Listen, TCP\_Open, TCP\_ReceiveBLOB, TCP\_Send.

TCP\_ReceiveBLOB (tcp\_ID; donnéesReçues) → Entier

Paramètre	Type		Description
tcp_ID	Entier long	→	Référence à une session TCP ouverte
donnéesReçues	BLOB	←	BLOB devant recevoir les données
Résultat	Entier	←	Code d'erreur

Description

La commande TCP\_ReceiveBLOB permet de recevoir des paquets de données au cours d'une session TCP. Son fonctionnement est semblable à celui de la commande TCP\_Receive, à la différence près qu'elle reçoit des données de type BLOB au lieu d'un texte. Ce principe permet de s'affranchir de la limite des 32 000 caractères inhérente aux données de type Texte, et de recevoir des données binaires.

Passez dans tcp\_ID la référence d'une session TCP ouverte avec la commande TCP\_Open ou TCP\_Listen.

Le paramètre donnéesReçues retourne les données reçues dans un BLOB. Lors de la réception de données par paquets TCP, vous ne pouvez pas avoir la certitude que la totalité des données ont été reçues via un seul appel à TCP\_ReceiveBLOB. La commande TCP\_ReceiveBLOB est donc généralement utilisée dans une boucle du type Repeter...Jusque qui vérifie en permanence le statut de la connexion ou attend une valeur particulière.

Exemple

Voici la structure type d'une méthode utilisant la commande TCP\_ReceiveBLOB :

```
C_BLOB($Blob_Reçu;$Blob_Concaténé)
C_ENTIER LONG($srcpos;$dstpos)
Repeter
⇒   $Err:=TCP_ReceiveBLOB ($TCP_ID;$Blob_Reçu)
    $Err:=TCP_State ($TCP_ID;$State)
    $srcpos:=0
    $dstpos:=Taille BLOB($Blob_Concaténé)
        `Concaténation des données reçues
    COPIER BLOB($Blob_Reçu;$Blob_Concaténé;$srcpos;$dstpos;
                                                Taille BLOB($Blob_Reçu))
Jusque (($State=0) | ($Err#0))
```

Référence

TCP\_Listen, TCP\_Open, TCP\_SendBLOB, TCP\_Receive.

TCP\_State (tcp\_ID; codeStatut) → Entier

Paramètre	Type		Description
tcp_ID	Entier long	→	Référence d'une session TCP ouverte
codeStatut	Entier	←	Code du statut TCP
Résultat	Entier	←	Code d'erreur

### Description

La commande TCP\_State renvoie une valeur indiquant le statut d'une connexion TCP particulière. tcp\_ID contient la référence d'une session TCP ouverte avec la commande TCP\_Open ou TCP\_Listen.

Le paramètre codeStatut retourne l'un des codes de statut suivants :

- 0 Connexion fermée
- 2 A l'écoute d'une connexion entrante
- 8 Connexion établie

### Exemple

Cet exemple suppose qu'une connexion TCP valide a été établie et est identifiée par la variable *\$tcp\_id*. Dans cet exemple, une commande est envoyée à un serveur Web pour demander une page d'informations, et les résultats sont récupérés dans une boucle de type Repeter. Comme les serveurs Web referment automatiquement les connexions dès qu'ils ont terminé leur tâche, la méthode continuera à recevoir des données jusqu'à ce que la connexion soit stoppée ou qu'une erreur se produise.

```

C_ENTIER LONG($tcp_id)
C_ENTIER(vEtat;$err)
C_TEXTE($commande;$tampon;$réponse)
Si (TCP_Send ($tcp_id;$commande)=0)
  vEtat:=0
  Repeter
    $err:=TCP_Receive ($tcp_id;$tampon)
⇒    $err:=TCP_State ($tcp_id;vEtat)
    $réponse:=$réponse+$tampon
  Jusque ((vState=0) | ($err#0))
Fin de si

```

### Référence

TCP\_Listen, TCP\_Open.

TCP\_Close (tcp\_ID) → Entier

Paramètre	Type		Description
tcp_ID	Entier long	→	Référence d'une session TCP ouverte
		←	0 = la session a été correctement refermée
Résultat	Entier	←	Code d'erreur

### Description

La commande TCP\_Close referme la session TCP référencée par tcp\_ID.

Lorsqu'une session TCP est ouverte, elle occupe l'un des 64 "canaux" (*streams*) disponibles. Si les sessions ne sont jamais refermées, lorsque leur nombre atteint 64 l'utilisateur ne peut plus en ouvrir d'autre.

tcp\_ID contient la référence d'une session TCP ouverte avec la commande TCP\_Open ou TCP\_Listen. Après exécution de la commande, si la session a été correctement refermée, ce paramètre prend la valeur 0 (zéro).

### Référence

TCP\_Listen, TCP\_Open.

# 7

---

## IT Internet



Les commandes de ce thème permettent d'effectuer des tâches courantes sur Internet. A l'aide de ces commandes, vous pourrez ainsi sonder (*Ping*) et explorer (*Finger*) une machine, obtenir l'heure d'une horloge réseau, décoder un nom de domaine ou une adresse IP, ou encore les convertir en valeurs de type entier long. Ces commandes s'utilisent généralement en association avec les autres commandes Internet de 4D.

NET\_Finger (nomServeur; texteRecherche; informations) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur
texteRecherche	Alpha	→	Texte de la recherche
informations	Texte	←	Résultats de la recherche
Résultat	Entier	←	Code d'erreur

### Description

La commande NET\_Finger permet d'obtenir des informations sur un compte utilisateur enregistré sur un serveur. La commande Unix *Finger* retourne l'heure de la dernière connexion d'un utilisateur ainsi que des informations supplémentaires que l'utilisateur choisit de fournir dans ses fichiers ".plan" et ".project".

NET\_Finger interroge, sur la machine désignée par nomServeur, le compte d'utilisateur spécifié par texteRecherche et retourne le résultat dans le paramètre informations.

Une recherche Finger peut être effectuée de deux manières :

- **Directement** : la recherche est effectuée directement sur la machine de l'utilisateur. Par exemple, pour obtenir des informations sur "johnt" chez "4d.com", vous pouvez écrire :

⇒ \$erreur:=**NET\_Finger** ("www.4d.com";"johnt";\$textefinger)

- **Indirectement** : dans ce cas, vous demandez à un serveur distant (qui accepte la commande *Finger*) d'effectuer la requête. Par exemple, la requête suivante demande à la machine identifiée par le nom de domaine "4d.com" d'effectuer une recherche *Finger* de l'utilisateur "johnt@4d.com".

⇒ \$erreur:=**NET\_Finger** ("www.4d.com";"johnt@4d.com";\$textefinger)

Bien que les principales informations renvoyées dans chaque cas soient globalement identiques, de légères différences peuvent être constatées. En effet, les paramétrages d'exécution de la commande Finger peuvent différer d'une machine à l'autre. En outre, des différences de formatage peuvent apparaître entre les résultats d'une commande Finger directe et ceux d'une commande indirecte, les recherches indirectes contenant souvent des retours à la ligne (*line feed*) supplémentaires.



Le paramètre `nomServeur` contient le nom ou l'adresse IP du serveur sur lequel l'utilisateur identifié par `texteRecherche` a un compte.

Le paramètre `texteRecherche` contient soit le texte à rechercher sur un serveur Finger, soit le nom ou l'adresse IP d'une machine. Si vous passez un nom d'utilisateur dans `texteRecherche`, la commande le recherchera dans le répertoire des noms d'utilisateurs du serveur. Si vous passez un nom de machine ou une adresse IP, la commande enverra la requête Finger à la machine spécifiée par l'intermédiaire du serveur Finger `nomServeur`.

Le paramètre `informations` retourne le résultat de la recherche.

NET\_Ping (nomServeur; texte; actif{; timeOut}) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur
texte	Texte	→	Texte à envoyer dans le "ping"
actif	Entier	←	1 = Actif, 0 = Timeout/Inactif
timeOut	Entier	→	Nombre de secondes d'attente, 0 = utiliser la valeur IT_SetTimeOut
Résultat	Entier	←	Code d'erreur

### Description

La commande NET\_Ping permet d'interroger une adresse IP distante pour vérifier qu'elle est active. Si la machine sondée exploite le protocole TCP/IP et si le réseau entre les deux sites est opérationnel, le statut 'Actif' est renvoyé.

Toute machine ayant une adresse IP accessible via le réseau mondial peut être sondée, y compris celles des utilisateurs finaux (toutefois, certains systèmes de sécurité appelés "firewalls" peuvent empêcher le sondage des machines placées sous leur protection).

Le paramètre nomServeur contient le nom de serveur ou l'adresse IP de la machine à sonder.

Le paramètre texte contient le texte à envoyer dans le "ping". Ce paramètre permet uniquement d'agir sur la taille du paquet TCP envoyé lors de l'exécution de la commande.

Le paramètre actif retourne un code indiquant l'état de la machine sondée. S'il reçoit la valeur 1, l'ordinateur est actif. S'il reçoit la valeur 0 (zéro), l'ordinateur est inactif ou la commande a dépassé le délai autorisé (*timeout*) sans recevoir de réponse.

Le paramètre timeOut spécifie le nombre maximum de secondes alloué pour l'exécution de la commande NET\_Ping. Si ce paramètre est omis ou si vous passez 0, la valeur de *timeout* spécifiée par la commande IT\_SetTimeOut sera utilisée.

### Référence

IT\_SetTimeOut.

NET\_Time (nomServeur; date; heure; décalage) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur NTP
date	Date	←	Date
heure	Entier long	←	Heure, exprimée en secondes depuis minuit
décalage	Entier	→	Nombre d'heures de décalage
Résultat	Entier	←	Code d'erreur

### Description

La commande NET\_Time permet de récupérer la date et l'heure courantes d'une horloge réseau sur Internet, et de leur appliquer le décalage nécessaire pour la conversion en heure locale de l'utilisateur.

**Note :** Cette commande n'affecte pas l'horloge interne de l'ordinateur.

Le paramètre nomServeur contient le nom ou l'adresse IP d'un serveur NTP (*Network Time Protocol*).

Le paramètre date retourne la date (au format date 4D) fournie par le serveur NTP et à laquelle le décalage a été appliqué.

Le paramètre heure retourne l'heure fournie par le serveur NTP, après l'application du décalage. Cette valeur représente le nombre de secondes écoulées depuis minuit à cette date. L'exemple suivant propose une méthode de conversion de cette valeur en une variable heure 4D.

Le paramètre décalage indique le nombre d'heures à ajouter ou à soustraire des valeurs reçues. Les horloges réseau d'Internet expriment leurs valeurs en temps universel (TU). Même si l'horloge réseau située est dans votre fuseau horaire, il est probable que vous deviez fournir une valeur de décalage pour compenser la différence entre votre heure locale et le temps universel.

## Exemple

L'exemple suivant récupère la date et l'heure de l'horloge réseau située sur le site "apple.com". La commande soustrait ensuite les sept heures de décalage spécifiées et renvoie la date et l'heure résultantes (l'heure est exprimée sous forme d'un entier long, qui peut ensuite être converti à l'aide de la commande 4D Chaîne heure, comme illustré ci-dessous).

```
C_DATE(vDateNet)
C_ENTIER LONG(vHeureNet)
C_HEURE(vHeure)
C_ENTIER(vDécalage)
⇒ Si (VérifErreur ("Heure_Net";NET_Time ("www.apple.com"; vDateNet; vHeureNet; -7)))
    vHeure:=Heure(Chaîne heure(vHeureNet)) `Convertit l'entier long en heure 4D
Fin de si
```

**Note :** La méthode *VérifErreur* est détaillée dans la description de la commande IT\_ErrorText.

## Référence

Chaîne heure.

NET\_NameToAddr (nomServeur; ip\_EntierLong) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur
ip_EntierLong	Entier long	←	Référence à l'adresse
Résultat	Entier	←	Code d'erreur

### Description

La commande NET\_NameToAddr retourne un entier long qui référence de manière unique le nom ou l'adresse IP d'un serveur.

**Important :** Afin d'assurer la compatibilité future de vos applications avec IPV6, l'utilisation de cette commande est déconseillée. Il est préférable d'utiliser NET\_Resolve qui travaille sur la base de chaînes de caractères.

Le paramètre nomServeur contient le nom ou l'adresse IP du serveur.

Le paramètre ip\_EntierLong retourne une valeur identifiant l'adresse IP spécifiée dans le paramètre nomServeur. Toute chaîne d'adresse IP peut être convertie en entier long.

La commande NET\_NameToAddr est utile pour économiser de la place lors du stockage des données. En effet, les valeurs stockées dans le format entier long sont plus compactes que les chaînes de caractères.

### Référence

NET\_AddrToName.

NET\_AddrToName (ip\_EntierLong; nomServeur; adresse\_IP) → Entier

Paramètre	Type		Description
ip_EntierLong	Entier long	→	Référence d'adresse
nomServeur	Alpha	←	Nom du serveur
adresse_IP	Alpha	←	Adresse IP
Résultat	Entier	←	Code d'erreur

### Description

La commande NET\_AddrToName retourne le nom et l'adresse IP d'un serveur à partir de sa référence unique ip\_EntierLong, générée par la commande NET\_NameToAddr.

**Important** : Afin d'assurer la compatibilité future de vos applications avec IPV6, l'utilisation de cette commande est déconseillée. Il est préférable d'utiliser NET\_Resolve qui travaille sur la base de chaînes de caractères.

ip\_EntierLong indique la référence unique d'une adresse IP.

nomServeur retourne le nom du serveur sous forme de chaîne de caractères.

adresse\_IP retourne l'adresse IP du serveur sous forme de chaîne de caractères. Si le nom du serveur ne peut être converti, adresse\_IP retourne une chaîne vide, aucune erreur n'est renvoyée.

### Référence

NET\_NameToAddr.

NET\_Resolve (nomServeur; ipOuServeur) → Entier

Paramètre	Type		Description
nomServeur	Alpha	→	Nom ou adresse IP du serveur
ipOuServeur	Alpha	←	Renvoie la valeur opposée
Résultat	Entier	←	Code d'erreur

Description

La commande NET\_Resolve "décode" le nom ou l'adresse IP d'un serveur.

- Si vous passez un nom de serveur dans le paramètre nomServeur, la commande NET\_Resolve renvoie l'adresse IP dans le paramètre ipOuServeur.
- Si vous passez une adresse IP dans le paramètre nomServeur, la commande NET\_Resolve renvoie le nom de serveur enregistré pour cette machine dans le paramètre ipOuServeur.

Exemple

La méthode suivante fournit d'abord un nom de serveur "www.netcom.com" à la commande NET\_Resolve afin d'obtenir son adresse IP. La méthode effectue ensuite un autre appel à la commande, en lui fournissant l'adresse IP afin d'obtenir son nom de serveur enregistré.

```
C_BOOLEAN($ERR)
C_ALPHA(80;$Converti) `Peut être une chaîne de caractères ou un texte de toute taille
⇒ $ERR:=VérifErr ("Convertir_NET";NET_Resolve ("www.netcom.com";$Converti))
  ` $Converti a renvoyé la valeur '192.100.81.100'
⇒ $ERR:=VérifErr ("Convertir_NET";NET_Resolve ($Converti;$Converti))
  ` $Converti a renvoyé la valeur 'www.netcom.com'
```

**Note :** La méthode *VérifErreur* est détaillée dans la description de la commande IT\_ErrorText.





# 8

---

## IT Utilities



Cette section regroupe un ensemble de routines utilitaires. Certaines de ces commandes permettent de déterminer l'environnement dans lequel fonctionnent les machines des utilisateurs, les versions des logiciels utilisés ou l'état courant et l'adresse IP de leur ordinateur.

Les autres commandes permettent de traiter les codes d'erreur, d'encoder et de décoder des fichiers, et d'agir sur la valeur d'attente maximale (*timeout*) par défaut applicable à la plupart des commandes des autres thèmes.

IT\_MacTCPInit → Entier

Paramètre	Type	Description
Cette commande ne requiert pas de paramètre		
Résultat	Entier	← Code d'erreur

### Description

La commande IT\_MacTCPInit initialise le gestionnaire TCP à utiliser avec les commandes Internet de 4D. La commande renvoie une erreur si le gestionnaire TCP ne peut pas être ouvert.

Il est recommandé de placer cette commande dans la Méthode base Sur ouverture des bases de données 4e Dimension. La commande ne doit être exécutée qu'une fois.

Si votre configuration TCP utilise un système d'accès réseau à distance tel que PPP ou SLIP, il est préférable de retarder l'exécution de IT\_MacTCPInit jusqu'à ce qu'une connexion TCP soit nécessaire, car cette commande provoque l'ouverture et l'initialisation de la connexion distante.

**Note :** La commande IT\_MacTCPInit doit être exécutée avant toute autre commande Internet de 4D (exception faite des commandes relatives au serveur Web 4D).

IT\_Platform → Entier

Paramètre	Type	Description
Cette commande ne requiert pas de paramètre		
Résultat	Entier	← Type de plate-forme (0 = MacOS 68K, 1 = MacOS PPC, 2 = Windows)

### Description

La fonction IT\_Platform retourne un entier indiquant la plate-forme en cours d'exploitation.

La fonction retourne 0 (zéro) pour la plate-forme MacOS 68000, 1 pour MacOS PowerPC et 2 pour Windows.

### Exemple

```
C_BOOLEEN (⌘ITnative)
⇒ ⌘ITnative:=(IT_Platform=1)
```

IT\_Version → Alpha

Paramètre	Type	Description
Cette commande ne requiert pas de paramètre		
Résultat	Alpha	← Chaîne de la version

### Description

La fonction IT\_Version renvoie une chaîne de caractères indiquant le numéro de version du plug-in 4D Internet Commands.

### Exemple

L'exemple suivant affiche une boîte de dialogue d'alerte indiquant la version de 4D Internet Commands utilisée :

⇒ **ALERTE**("4D Internet Commands, version : "+*IT\_Version*)

IT\_TCPversion (typePile; versionPile) → Entier

Paramètre	Type		Description
typePile	Entier	←	0 = Aucun, 1 = MacTCP, 2 = Open Transport, 3 = WinSock
versionPile	Texte	←	Numéro de version de la pile TCP
Résultat	Entier	←	Code d'erreur

### Description

La commande IT\_TCPversion retourne des informations sur le type de pile TCP en cours d'utilisation avec les commandes Internet de 4D. Le type de pile varie selon la plate-forme. Sur Macintosh, MacTCP et Open Transport sont tous deux pris en charge. Sous Windows, la pile TCP WinSock est prise en charge.

Le paramètre typePile retourne un code exprimant le type de pile TCP utilisé, en fonction du tableau suivant :

Code	Pile TCP
0	Aucun
1	MacTCP
2	Open Transport
3	WinSock

Le paramètre versionPile retourne le numéro de version de la pile TCP utilisée.

IT\_MacTCPVer (codeVersion) → Entier

Paramètre	Type		Description
codeVersion	Entier	←	Code de la version de MacTCP installée
Résultat	Entier	←	Code d'erreur

### Description

La commande IT\_MacTCPVer est devenue obsolète avec l'ajout de la commande IT\_TCPversion qui permet d'obtenir des informations plus complètes sur MacTCP, Open Transport et Winsock.

Cette commande renvoie dans le paramètre codeVersion la version de MacTCP installée sur l'ordinateur de l'utilisateur. L'entier renvoyé provient du Gestalt Manager d'Apple et ne correspond pas directement au numéro de version de MacTCP. Cette commande est utile pour déterminer si un utilisateur exécute la version minimale de MacTCP, requise pour l'utilisation des commandes Internet de 4D.

Actuellement, le gestionnaire Gestalt renvoie les valeurs suivantes correspondant aux versions MacTCP ci-dessous. A mesure qu'Apple publie des mises à jour de MacOS et de MacTCP, de nouvelles valeurs pourront être renvoyées.

### Codes de version

0 = Gestionnaire MacTCP non ouvert

1 = v1.1

2 = v1.1.1

3 = v2.0



## Exemple

La fonction suivante peut être appelée au démarrage pour mettre à Vrai ou à Faux une variable interprocess indiquant si l'utilisateur dispose d'une version suffisante de MacTCP.

```
`Méthode VERIFIER_TCP ( {»Version} ) : Booléen [MacTCP installé]
C_ENTIER($ERR)
C_BOOLEEN($TCP_OK)
C_ENTIER LONG(vVersionMacTCP)
⇒ $ERR:=IT_MacTCPVer (vVersionMacTCP)
   $TCP_OK:=((($ERR=0) & (vVersionMacTCP#0)) `Vrai = Pas d'erreur, la version MacTCP
   `est connue

Au cas ou
: (Non($TCP_OK) & (¶AFFICHERERREURS))
   `Quelque chose ne va pas, afficher l'erreur
   ALERTE("Problème à l'ouverture de votre gestionnaire MacTCP. Vérifiez qu'il est
   correctement installé.")
: ((($TCP_OK) & (Nombre de parametres=1)) `Tout va bien, retourner le numéro
   `de version
   ENUMERATION VERS TABLEAU("Versions MacTCP";$aTCPversion)
   Si (vTCPversion<=Taille tableau($aVersionTCP))
       $1->:=$aVersionTCP{vVersionMacTCP}
   Sinon
       $1->:="Inconnue"
   Fin de si
Fin de cas
$0:=$TCP_OK `Renvoyer Vrai si la vérification de la version est correcte
```

IT\_MyTCPAddr (adresse\_IP; sousRéseau) → Entier

Paramètre	Type		Description
adresse_IP	Alpha	←	Adresse IP de la machine de l'utilisateur
sousRéseau	Alpha	←	Masque de sous-réseau au format IP
Résultat	Entier	←	Code d'erreur

**Description**

La commande IT\_MyTCPAddr renvoie l'adresse IP de la machine qui exécute la commande.

Le paramètre adresse\_IP retourne une chaîne de caractères contenant l'adresse IP.

Le paramètre sousRéseau retourne une chaîne de caractères contenant le masque de sous-réseau de l'adresse IP.

IT\_SetTimeout (timeOut) → Entier

Paramètre	Type		Description
timeOut	Entier	→	Délai d'attente en secondes (0 à 127)
Résultat	Entier	←	Code d'erreur

### Description

La commande IT\_SetTimeout vous permet de définir la valeur du délai d'attente maximal (*Timeout*) alloué par défaut pour l'exécution des commandes Internet de 4D. Cette valeur doit être exprimée en secondes et être comprise entre 0 et 127.

Passez dans timeOut la valeur souhaitée. Les commandes suivantes sont affectées par IT\_SetTimeout :

TCP_Open	POP3_Login
FTP_Login	POP3_BoxInfo
FTP_Append	POP3_Delete
FTP_Send	POP3_Reset
FTP_Receive	POP3_MsgInfo
SMTP_QuickSend	POP3_MsgLstInfo
SMTP_Send	POP3_GetMessage
NET_Finger	POP3_MsgLst
NET_Ping	POP3_Download
NET_Time	POP3_VerifyID
	POP3_UIDToNum

**Note :** La définition du timeOut à 0 (zéro) pour la commande TCP\_Listen lui permet d'être à l'écoute indéfiniment. Veillez à passer une autre valeur après avoir utilisé cette commande, la valeur zéro signifiant généralement l'utilisation de la valeur par défaut. De plus, cette valeur est également utilisée pour les "timeouts TCP/IP" ET le "timeout d'attente de réponse". Si vous fixez ce délai d'attente à zéro, aucune réponse n'aura le temps d'être reçue.

### Référence

IT\_GetTimeout.

IT\_GetTimeout (timeOut) → Entier

Paramètre	Type		Description
timeOut	Entier	←	Délai d'attente en secondes
Résultat	Entier	←	Code d'erreur

**Description**

La commande IT\_GetTimeout renvoie la valeur courante du délai d'attente (*timeout*) pour les commandes énumérées dans la description de la commande IT\_SetTimeout.

Le paramètre timeOut retourne la valeur courante du délai d'attente en secondes.

**Référence**

IT\_SetTimeout.

IT\_ErrorText (erreur) → Alpha

Paramètre	Type		Description
erreur	Entier	→	Code d'erreur
Résultat	Alpha	←	Texte de l'erreur

### Description

La commande IT\_ErrorText retourne le libellé de l'erreur dont le numéro a été passé en paramètre. Notez qu'il s'agit d'une des rares commandes Internet de 4D qui ne retourne pas un entier comme valeur.

Le paramètre erreur contient le numéro de l'erreur.

### Exemple

La méthode *VérifErreur* suivante affiche un message d'alerte expliquant l'erreur. Cette méthode reçoit deux paramètres : le nom de la commande (\$Commande) et la valeur de l'erreur (fournie par l'exécution de la commande dans le paramètre de la méthode).

*VérifErreur* renvoie une valeur booléenne indiquant si la commande a retourné le code d'erreur 0 (zéro). Si ce n'est pas le cas, la valeur retournée (\$0) est Faux, sinon Vrai.

```

    `Méthode VérifErreur ("NOM DE COMMANDE"; Erreur# ) -> Vrai/Faux
    C_TEXTE(vMsgErreur)
    $Commande:=$1
    $Erreur:=$2
    $Resultat:=Vrai
    Si ($Erreur#0)
        $Resultat:=Faux
⇒    vMsgErreur:=IT_ErrorText ($Erreur)
        ALERTE("ERREUR -- "+Caractere(13)+"Commande: "+$Commande+Caractere(13)
            +"Code d'erreur:"+Chaine($Erreur)+Caractere(13)
            +"Description: "+vMsgErreur)
    Fin de si
    $0:=$Resultat

```

### Référence

Annexe A, Conseils de programmation.

IT\_Encode (nomFichier; nomFichierCodé; modeCodage) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Chemin d'accès local à un fichier
nomFichierCodé	Texte	→	Chemin d'accès du local encodé
		←	Chemin d'accès au fichier encodé résultant
modeCodage	Entier	→	1 = BinHex 2 = Base64 (Data fork uniquement) 3 = AppleSingle 4 = AppleDouble 5 = AppleSingle ET Base64 6 = AppleDouble ET Base64 7 = UUEncode 8 = MacBinary
Résultat	Entier	←	Code d'erreur

### Description

La commande IT\_Encode encode un fichier au moyen du modeCodage spécifié. Le fichier désigné n'est pas modifié, seule une copie encodée est créée. Le nom du fichier encodé créé est le nom initial du fichier plus un suffixe ajouté pour spécifier la méthode de codage. Pour le codage Binhex, le suffixe ".hqx" est ajouté. Pour le codage Base64, le suffixe ".b64" est ajouté. Pour le codage AppleSingle, le suffixe ".as" est ajouté.

Le paramètre nomFichier contient le chemin d'accès complet au fichier que vous voulez encoder. Si vous passez une chaîne vide dans ce paramètre, une boîte de dialogue standard d'ouverture de documents apparaît, permettant à l'utilisateur de sélectionner lui-même le fichier.

Le paramètre nomFichierCodé peut contenir le chemin d'accès complet de la copie encodée du fichier, ce qui vous permet de définir son nom et son emplacement. Sinon, la commande IT\_Encode attribue un nom par défaut au document et le place dans le même dossier que le fichier désigné dans le premier paramètre. Qu'il soit spécifié ou non, le chemin d'accès final du document encodé est retourné dans le paramètre nomFichierCodé. En raison des conflits de noms potentiels dans le répertoire spécifié, il est conseillé de prendre en compte la valeur résultante du paramètre.

**Note :** Si le nom du fichier initial plus le suffixe ajouté comportent plus de 31 caractères, l'erreur -37 est générée. Vous devez donc vous assurer que le nom du fichier initial plus le suffixe ne dépassent pas 31 caractères.

Le paramètre `modeCodage` vous permet de définir la méthode d'encodage à appliquer au fichier. La valeur par défaut est 1 (encodage binhex). Les méthodes d'encodage proposées sont les suivantes :

<b>Code</b>	<b>Méthode</b>
1	BinHex
2	Base64 (Data fork uniquement)
3	AppleSingle
4	AppleDouble
5	AppleSingle et Base64
6	AppleDouble et Base64
7	UUEncode
8	MacBinary

Pour le codage au moyen d'AppleDouble (modes de codage 4 & 6), deux fichiers sont créés, appelés "%nomFichier" et "nomFichier".

### **Référence**

IT\_Decompile.

IT\_Decode (nomFichier; nomFichierDécodé; modeDécodage) → Entier

Paramètre	Type		Description
nomFichier	Texte	→	Chemin d'accès local à un fichier encodé
nomFichierDécodé	Texte	→	Chemin d'accès du fichier décodé
		←	Chemin d'accès résultant du fichier décodé
modeDécodage	Entier	→	1 = BinHex 2 = Base64 (Data fork uniquement) 3 = AppleSingle 4 = AppleDouble 5 = AppleSingle ET Base64 6 = AppleDouble ET Base64 7 = UUEncode 8 = MacBinary
Résultat	Entier	←	Code d'erreur

### Description

La commande IT\_Decode décode un fichier au moyen du modeDécodage spécifié. Le fichier original n'est pas modifié, une copie décodée est créée.

Le paramètre nomFichier contient le chemin d'accès complet du fichier que vous voulez décoder. Si vous passez une chaîne vide dans ce paramètre, une boîte de dialogue standard d'ouverture de documents apparaît, permettant à l'utilisateur de sélectionner lui-même le fichier.

Le paramètre nomFichierDécodé peut contenir le chemin d'accès complet de la copie décodée du fichier, ce qui vous permet de définir son nom et son emplacement. Sinon, la commande IT\_Decode attribue un nom par défaut au document et le place dans le même répertoire que le fichier spécifié dans le premier paramètre. Qu'il soit spécifié ou non, le chemin d'accès final du document décodé est retourné dans le paramètre nomFichierDécodé.

**Note :** Si le nom du fichier initial plus le suffixe ajouté comportent plus de 31 caractères, la commande tronque le nom du fichier initial pour définir le nom de fichier encodé.



Le paramètre `modeDécodage` vous permet d'indiquer la méthode de décodage à appliquer au fichier. La valeur par défaut est 1 (décodage binhex). Les méthodes de décodage proposées sont les suivantes :

<b>Code</b>	<b>Méthode</b>
-------------	----------------

1	BinHex
2	Base64 (Data fork uniquement)
3	AppleSingle
4	AppleDouble
5	AppleSingle et Base64
6	AppleDouble et Base64
7	UUEncode
8	MacBinary

Pour le décodage au moyen d'AppleDouble (modes de décodage 4 & 6), la commande recherche la partie ressources dans un fichier nommé "%nomFichier".

### Référence

FTP\_MacBinary, FTP\_Receive, IT\_Encode.

---

IT\_GetProxy (protocole; typeProxy; nomServeurProxy; portProxy; idUtilisateurProxy) → Entier

Paramètre	Type		Description
protocole	Entier	→	1 = FTP ; 2 = SMTP ; 3 = POP3
typeProxy	Entier	←	0 = Aucun ; 1 = SOCKS
nomServeurProxy	Alpha	←	Nom ou adresse IP du serveur proxy SOCKS
portProxy	Entier	←	Port du proxy auquel se connecter
idUtilisateurProxy	Texte	←	ID d'utilisateur pour SOCKS
Résultat	Entier	←	Code d'erreur

### Description

La commande IT\_GetProxy retourne les paramètres courants appliqués au routage, via un serveur proxy SOCKS, du protocole spécifié. Les valeurs sont celles par défaut à moins qu'un appel antérieur à IT\_SetProxy ne les ait modifiées. Pour plus d'informations sur ces paramètres, reportez-vous à la description de la commande IT\_SetProxy.

Le paramètre protocole vous permet de spécifier le protocole à filtrer. Passez 1 pour le protocole FTP, 2 pour le protocole SMTP ou 3 pour le protocole POP3.

Le paramètre typeProxy retourne un code indiquant si un serveur proxy SOCKS est utilisé. La valeur 1 signifie que toutes les requêtes pour le protocole spécifié transitent par un serveur SOCKS. La valeur 0 (zéro) signifie que les requêtes ne transitent pas par un quelconque serveur SOCKS.

Le paramètre nomServeurProxy retourne le nom ou l'adresse IP du serveur proxy SOCKS utilisé.

Le paramètre portProxy retourne le numéro du port utilisé avec le protocole spécifié pour communiquer avec le serveur proxy SOCKS.

Le paramètre idUtilisateurProxy retourne l'ID de l'utilisateur.

### Référence

IT\_SetProxy.

---

IT\_SetProxy (protocole; typeProxy; nomServeurProxy; portProxy; idUtilisateurProxy) → Entier

Paramètre	Type		Description
protocole	Entier	→	1 = FTP ; 2 = SMTP ; 3 = POP3
typeProxy	Entier	→	0 = Aucun ; 1 = SOCKS
nomServeurProxy	Alpha	→	Nom ou adresse IP du serveur proxy SOCKS
portProxy	Entier	→	Port du proxy auquel se connecter
idUtilisateurProxy	Texte	→	ID d'utilisateur pour SOCKS
Résultat	Entier	←	Code d'erreur

### Description

La commande IT\_SetProxy permet d'ouvrir une connexion au moyen du protocole spécifié, puis d'envoyer toutes les requêtes proposées au protocole spécifié via le serveur SOCKS (Proxy SOCKS). Si vous vous connectez uniquement en Intranet, vous n'aurez généralement pas à passer par le serveur SOCKS, sauf si le paramétrage du *firewall* ("pare feu") de votre réseau le requiert. La commande IT\_SetProxy a une portée interprocess et s'applique à toutes les sessions de communication utilisant le protocole spécifié, quel que soit le process 4D.

**Note :** Socks (ou "SOCKS") est un protocole utilisable par un serveur proxy. Il permet de filtrer les requêtes des utilisateurs du réseau d'une entreprise, lorsque celles-ci doivent être transmises sur Internet. Si votre poste de travail se trouve derrière un *firewall* et si vous souhaitez accéder à des informations sur Internet, le serveur SOCKS reçoit votre requête, la transmet à travers le *firewall*, puis récupère et renvoie les informations à votre application cliente.

Le paramètre protocole définit le protocole qui doit être filtré par le serveur proxy SOCKS spécifié. Passez 1 pour désigner le protocole FTP, 2 pour le protocole SMTP et 3 pour le protocole POP3.

Le paramètre typeProxy indique si le protocole spécifié doit ou non être filtré par un serveur proxy SOCKS. Passez 1 pour faire transiter toutes les requêtes du protocole spécifié par le serveur SOCKS spécifié, sinon passez 0.

Le paramètre nomServeurProxy contient le nom ou l'adresse IP du serveur proxy SOCKS.

Le paramètre portProxy contient le port à utiliser pour que le protocole spécifié communique avec le serveur proxy SOCKS.

Le paramètre idUtilisateurProxy identifie l'utilisateur. L'ID d'utilisateur est attribué par votre administrateur réseau. idUtilisateurProxy peut être un texte vide ("").

### Exemple

La méthode suivante permet d'acheminer toutes les connexions FTP via le serveur proxy SOCKS spécifié.

```
⇒ $err=IT_SetProxy (1;1;$ajouterProxy;$PortProxy;"") `Proxy SOCKS FTP
   $err=FTP_Login ("ftp.4d.com";"anonymous";"dbody@aol.com";$ftpID)
   $err=FTP_GetFileInfo ($ftpID;$vchemin;$vtaille;$vdateModif)
   $err=FTP_Receive ($ftpID;$vchemin;"";0)
   $err=FTP_Logout ($ftpID)
```

**Note :** Par souci de clarté, cet exemple ne contient pas de vérification d'erreur.

L'instruction suivante permet de ne plus passer par le serveur proxy SOCKS lors des requêtes FTP.

```
⇒ $err=IT_SetProxy (1;0;$ajouterProxy;$portProxy;"")
```

### Référence

IT\_GetProxy.

IT\_GetPort (protocole; port) → Entier

Paramètre	Type		Description
protocole	Entier	→	1 = FTP ; 2 = SMTP ; 3 = POP3
port	Entier	←	Numéro de port
Résultat	Entier	←	Code d'erreur

**Description**

Pour le protocole spécifié, la commande IT\_GetPort récupère le numéro du port courant utilisé par les commandes Internet de 4D relatives à ce protocole.

**Référence**

Annexe B, Numéros des ports TCP, IT\_SetPort.

IT\_SetPort (protocole; port) → Entier

Paramètre	Type		Description
protocole	Entier	→	1 = FTP ; 2 = SMT P; 3 = POP3
port	Entier	→	Numéro de port
Résultat	Entier	←	Code d'erreur

**Description**

Pour le protocole spécifié, la commande IT\_SetPort dirige toutes les communications ultérieures utilisant ce protocole vers le port spécifié.

**Référence**

Annexe B, Numéros des ports TCP, IT\_GetPort.

# 9

---

## Annexes





**Exécution des commandes dans des boucles 'Au cas ou'**

Dans de nombreux exemples de ce manuel, on utilise une structure de programmation particulière. Ces exemples exécutent des séries de commandes en utilisant l'instruction **Au cas ou** de manière peu orthodoxe.

En fait, de nombreuses fonctionnalités des commandes Internet de 4D Internet requièrent l'exécution complète d'une séquence de commandes. L'échec d'une seule commande de la séquence étant suffisant pour interrompre la poursuite du processus, l'utilisation de **Si** imbriqués sur de nombreux niveaux serait fastidieux :

```

Si (SMTP_New($smtp_id)=0)
  Si (SMTP_Host ($smtp_id;␣pref_Serveur)=0)
    Si (SMTP_From ($smtp_id;vDe)=0)
      Si (SMTP_To ($smtp_id;vA)=0)
        ...etc, etc...
      Fin de si
    Fin de si
  Fin de si
Fin de si

```

La solution employée dans ce manuel consiste à s'appuyer sur la manière dont 4D exécute les instructions **Au cas ou**. Chaque condition d'une instruction **Au cas ou** est exécutée par 4D pour déterminer si la valeur renvoyée est **Vraie** ou **Fausse**. Lorsque chaque condition d'une instruction **Au cas ou** renvoie une valeur fausse, toutes les conditions sont donc exécutées. Les lignes suivantes remplacent le code décrit plus haut :

```

$EnvoyeOK:=Faux `Ce "drapeau" indique si tous les appels ont été transmis
Au cas ou
  : (SMTP_New ($smtp_id)#0)
  : (SMTP_Host ($smtp_id;␣pref_Serveur)#0)
  : (SMTP_From ($smtp_id;vDe)#0)
  : (SMTP_To ($smtp_id;vA)#0)
  : (SMTP_Subject ($smtp_id;vSujet)#0)
  : (SMTP_Body ($smtp_id;vMessage)#0)
  : (SMTP_Send ($smtp_id)#0)
Sinon
  $EnvoyeOK:=Vrai `Le message a été composé et envoyé
Fin de cas

```

```
Si ($smtp_id#0) `Si un message a été créé en mémoire, nous devons l'effacer
$OK:=SMTP_Clear ($smtp_id)
Fin de si
```

Dans cet exemple, chaque commande renvoie l'erreur 0 (zéro) si son exécution s'est déroulée correctement. Pour pouvoir évaluer chaque condition, 4D doit en fait exécuter chaque ligne. Comme chaque condition compare le résultat à "différent de zéro", la valeur renvoyée est toujours **Fausse** et 4e Dimension ne trouve pas de condition à laquelle s'arrêter jusqu'à ce que l'une des commandes échoue. Si chaque commande est exécutée correctement, 4D poursuit l'exécution de la méthode jusqu'à la condition **Si** dans laquelle le drapeau *\$EnvoyeOK* prend la valeur **Vraie** afin d'indiquer que le message a été composé et envoyé sans incident.

### Recommandations pour une réponse automatique à un courrier POP3

Si vous envisagez de mettre en place, à l'intérieur de votre base de données, un système de messagerie permettant notamment aux utilisateurs de "répondre" aux courriers qu'ils ont reçu, voici quelques recommandations tirées de la RFC 822 :

- L'adresse électronique de l'"Expéditeur" (*Sender*) ne doit recevoir que des réponses relatives à des problèmes de distribution de courrier et **non** des réponses se rapportant au sujet traité dans le message. En l'absence d'en-tête "Expéditeur" (*Sender*), les problèmes doivent être envoyés à l'adresse figurant dans l'en-tête "Emetteur(s)" (*From*).
- L'adresse de l'"Expéditeur" (*Sender*) ne doit jamais être utilisée dans le cadre de l'utilisation d'un automate. L'automate devra plutôt utiliser l'en-tête "Réponse à" (*Reply-To*) ou "Emetteur(s)" (*From*) pour répondre à un message, le choix de l'en-tête le plus approprié dépendant des facteurs indiqués ci-dessous.
- Si l'en-tête "Réponse à" (*Reply-To*) existe et contient une ou plusieurs adresses, les réponses doivent alors être adressées aux personnes de cette liste. Les adresses de l'en-tête "Réponse à" (*Reply-To*) ont priorité sur celles figurant dans l'en-tête "Emetteur(s)" (*From*). En l'absence d'en-tête "Réponse à" mais en présence d'un en-tête "Emetteur(s)", les réponses doivent être envoyées aux adresses indiquées dans ce dernier.

Ces recommandations ont pour seul but de faciliter la prise de décision lorsque l'adressage du courrier est géré par programmation dans le cas d'actions de type "Réponse". Une fois le message de réponse créé, l'utilisateur final peut parfaitement annuler ces paramètres par défaut avant d'envoyer le message.

**Choix du numéro de port**

- 0 à 1023 (Ports réservés) : Ces ports sont affectés par l'I.A.N.A. (Internet Assigned Numbers Authority) et sur la plupart des systèmes ne peuvent être utilisés que par des process système (ou racine) ou par des programmes exécutés par des utilisateurs disposant de privilèges d'accès avancés.
  - 20 et 21 FTP;
  - 23 TELNET;
  - 25 SMTP;
  - 37 NTP;
  - 80 et 8080 HTTP.
- 1024 à 49151 (Ports enregistrés) : Ces ports sont enregistrés par l'I.A.N.A. et peuvent être utilisés sur la plupart des systèmes par des process utilisateurs ou par des programmes exécutés par des utilisateurs sans privilèges particuliers (routeurs, applications spécifiques...)
- 49152 à 65535 (Ports dynamiques et/ou privés) : Ces ports sont d'utilisation libre.

Les personnes souhaitant utiliser les commandes TCP/IP pour synchroniser des bases de données doivent utiliser des numéros de port supérieurs à 49151.

Pour de plus amples informations, veuillez visiter le site Web de l'I.A.N.A. :  
<http://www.iana.org>

**Numéros de port TCP**

daytime	13	Daytime
qotd	17	Quote of the Day
ftp-data	20	File Transfer [données par défaut]
ftp	21	File Transfer [contrôle]
telnet	23	Telnet
smtp	25	Simple Mail Transfer
time	37	Time
nicname	43	Who Is
domain	53	Domain Name Server
sql*net	66	Oracle SQL*NET
gopher	70	Gopher
finger	79	Finger
http	80	World Wide Web HTTP
poppassd	106	Password Server
rtelnet	107	Remote Telnet Service

pop2	109	Post Office Protocol - Version 2
pop3	110	Post Office Protocol - Version 3
sunrpc	111	SUN Remote Procedure Call
auth	113	Authentication Service
sftp	115	Simple File Transfer Protocol
sqlserv	118	SQL Services
nntp	119	Network News Transfer Protocol
ntp	123	Network Time Protocol
pwdgen	129	Password Generator Protocol
imap2	143	Interactive Mail Access Protocol v2
news	144	NewS
sql-net	150	SQL-NET
multiplex	171	Network Innovations Multiplex
cl/1	172	Network Innovations CL/1
at-rtmp	201	AppleTalk Routing Maintenance
at-nbp	202	AppleTalk Name Binding
at-3	203	AppleTalk Unused
at-echo	204	AppleTalk Echo
at-5	205	AppleTalk Unused
at-zis	206	AppleTalk Zone Information
at-7	207	AppleTalk Unused
at-8	208	AppleTalk Unused
ipx	213	IPX
netware-ip	396	Novell Netware sur IP
timbuktu	407	Timbuktu
conference	531	conversation
netnews	532	readnews
netwall	533	Pour émission d'urgence
uucp	540	uucpd
uucp-rlogin	541	uucp-rlogin
whoami	565	whoami
ipcserver	600	Sun IPC server
phonebook	767	téléphone
accessbuilder	888	AccessBuilder

Toutes les commandes Internet de 4D (à l'exception de IT\_ErrorText et IT\_Version) sont des fonctions retournant un entier comme résultat. Cet entier contient un numéro d'erreur que la commande doit retourner à la base de données 4D.

Si une commande aboutit, un zéro est renvoyé.

L'origine d'une erreur peut généralement être déterminée à partir de l'intervalle de valeurs dans lequel se situe son numéro. Le tableau suivant vous permet de distinguer les sources d'erreurs en fonction de leur numéro :

Numéro d'erreur	Générée par
Erreur < 0	Système d'exploitation
0	Pas d'erreur
Erreur >= 10000	Commandes Internet de 4D

### Codes d'erreurs de 4D Internet Commands

Si une erreur se produit durant l'exécution d'une commande Internet de 4D, une des valeurs suivantes sera retournée :

10000	user cancelled a dialog or progress.
10001	unimplemented Internet command.
10002	invalid array type.
10003	no more (TCP,SMTP,POP3, etc. ) references available.
10004	invalid reference.
10005	need a "Host" for use in the "SMTP_Send" command.
10006	need a "From" for use in the "SMTP_Send" command.
10007	need a recipient for use in the "SMTP_Send" command.
10008	already logged in.
10009	error trying to make a POP3 connection.
10010	error with POP3 USER.
10011	error with POP3 PASS.
10012	error with POP3 QUIT.
10013	error with POP3 STAT.
10014	error with POP3 LIST.
10015	error with POP3 UIDL.
10016	error with POP3 DELE.
10017	error with POP3 RSET.
10018	invalid message number.
10019	invalid character offset.
10020	invalid character length.
10021	error with POP3 RETR.
10022	field was not found in mail Header.
10023	no attachments found.
10024	error in processing BinHex.
10025	BinHex checksum error.

10026 Internet commands unavailable. Probably because MacTCP is not installed  
10027 Connection no longer exists  
10028 Exceeded 32k limit  
10029 Error with POP3 NOOP  
10030 POP3 session was closed by the server  
10031 Error with POP3 APOP  
10032 Unknown or invalid response.  
10033 SMTP 421 - Service not available, closing transmission channel.  
10034 SMTP 450 - Requested mail action not taken: mailbox unavailable.  
10035 SMTP 451 - Requested action aborted: local error in processing.  
10036 SMTP 452 - Requested action not taken: insufficient system storage.  
10037 SMTP 500 - Syntax error, command unrecognized.  
10038 SMTP 501 - Syntax error in parameters or arguments.  
10039 SMTP 502 - Command not implemented.  
10040 SMTP 503 - Bad sequence of commands.  
10041 SMTP 504 - Command parameter not implemented.  
10042 SMTP 550 - Requested action not taken: mailbox unavailable.  
10043 SMTP 551 - User not local; please try <forward-path>.  
10044 SMTP 552 - Requested mail action aborted: exceeded storage allocation.  
10045 SMTP 553 - Requested action not taken: mailbox name not allowed.  
10046 SMTP 554 - Transaction failed.  
10047 FTP 421 - Service not available, closing control connection.  
10048 FTP 425 - Can't open data connection.  
10049 FTP 426 - Connection closed; transfer aborted.  
10050 FTP 450 - Requested file action not taken. File unavailable (e.g.,file busy).  
10051 FTP 451 - Requested action aborted: local error in processing.  
10052 FTP 452 - Requested action not taken. Insufficient storage space in system.  
10053 FTP 500 - Syntax error, command unrecognized.  
10054 FTP 501 - Syntax error in parameters or arguments.  
10055 FTP 502 - Command not implemented.  
10056 FTP 503 - Bad sequence of commands.  
10057 FTP 504 - Command not implemented for that parameter.  
10058 FTP 530 - Not logged in.  
10059 FTP 532 - Need account for storing files.  
10060 FTP 550 - Requested action not taken. File unavailable (e.g., file not found, no access).  
10061 FTP 551 - Requested action aborted: page type unknown.  
10062 FTP 552 - Requested file action aborted. Exceeded storage allocation (for current directory or dataset).  
10063 FTP 553 - Requested action not taken. File name not allowed.  
10064 No response has been received within the given timeout period.  
10065 Not a FTP file.  
10066 Error in processing Base64.  
10067 Error in processing AppleSingle.  
10068 Error in processing Quoted-Printable.  
10069 FTP session was closed by the server.  
10070 Not a FTP directory.  
10071 TCP session was closed by the server

10072	Invalid encode kind
10073	Invalid decode kind
10074	An asynchronous DNR call did not complete
10075	An asynchronous OpenTransport call did not complete
10076	OpenTransport bind failed
10077	OpenTransport connect failed
10078	Maximum MacTCP streams reached
10079	Error in processing uuencode
10080	Cannot load ICMP library
10081	Error in processing MacBinary
10082	MacBinary checksum error
10083	Could not open a file
10084	No FTP information received
10085	Unknown FTP information received

### **Codes d'erreurs Open Transport**

-3211	Open Transport Out of Memory
-3201	Open Transport Not Found
-3216	Open Transport duplicate found
-3150	A Bad address was specified
-3151	A Bad option was specified
-3152	Missing access permission
-3153	Bad provider reference
-3154	No address was specified
-3155	Call issued in wrong state
-3156	Sequence specified does not exist
-3157	A system error occurred
-3158	An event occurred - call Look()
-3159	An illegal amount of data was specified
-3160	Passed buffer not big enough
-3161	Provider is flow-controlled
-3162	No data available for reading
-3163	No disconnect indication available
-3164	No Unit Data Error indication available
-3165	A Bad flag value was supplied
-3166	No orderly release indication available
-3167	Command is not supported
-3168	State is changing - try again later
-3169	Bad structure type requested for OTAlloc
-3170	A bad endpoint name was supplied
-3171	A Bind to an in-use addr with qlen > 0
-3172	Address requested is already in use
-3173	Accept failed because of pending listen
-3174	Tried to accept on incompatible endpoint
-3175	kOTResQLenErr
-3176	kOTResAddressErr
-3177	kOTQFullErr

- 3178 An unspecified provider error occurred
- 3179 A synchronous call at interrupt time
- 3180 The command was cancelled
- 3200 Permission denied
- 3201 No such file or directory
- 3202 No such resource
- 3203 Interrupted system service
- 3204 I/O error
- 3205 No such device or address
- 3208 Bad file number
- 3210 Try operation again later
- 3211 Not enough space
- 3212 Permission denied
- 3213 Bad address
- 3215 Device or resource busy
- 3216 File exists
- 3218 No such device
- 3221 Invalid argument
- 3224 Not a character device
- 3231 Broken pipe
- 3233 Message size too large for STREAM
- 3234 Call would block, so was aborted
- 3234 or a deadlock would occur
- 3236 kEALREADYErr
- 3237 Socket operation on non-socket
- 3238 Destination address required
- 3239 Message too long
- 3240 Protocol wrong type for socket
- 3241 Protocol not available
- 3242 Protocol not supported
- 3243 Socket type not supported
- 3244 Operation not supported on socket
- 3247 Address already in use
- 3248 Can't assign requested address
- 3249 Network is down
- 3250 Network is unreachable
- 3251 Network dropped connection on reset
- 3252 Software caused connection abort
- 3253 Connection reset by peer
- 3254 No buffer space available
- 3255 Socket is already connected
- 3256 Socket is not connected
- 3257 Can't send after socket shutdown
- 3258 Too many references: can't splice
- 3259 Connection timed out
- 3260 Connection refused
- 3263 Host is down
- 3264 No route to host



-3269	kEPROTOErr
-3270	kETIMEErr
-3271	kENOSRErr
-3272	kEBADMSGErr
-3273	kECANCELErr
-3274	kENOSTRErr
-3275	kENODATAErr
-3276	kEINPROGRESSErr
-3277	kESRCHErr
-3278	kENOMSGErr
-3279	kOTClientNotInittedErr
-3280	kOTPortHasDiedErr
-3281	kOTPortWasEjectedErr
-3282	kOTBadConfigurationErr
-3283	kOTConfigurationChangedErr
-3284	kOTUserRequestedErr
-3285	kOTPortLostConnection

### Codes d'erreurs MacTCP & Divers

-33	Unable to write to the disk. The File Directory is full.
-34	Unable to write to the disk. The disk is full.
-35	No such volume.
-36	I/O error.
-37	Bad file name or volume name.
-38	File is not open for reading or writing.
-39	Unable to read from the file. The end of file was reached.
-42	Unable to continue because too many files are open.
-43	The file cannot be found.
-44	The volume is locked at the hardware level.
-45	The file is locked.
-46	The volume is locked at the software level.
-47	One or more files are already open by another application.
-48	A file with this name already exists.
-49	File already open with write permission.
-54	Permissions error on opened file.
-57	Not a Macintosh volume.
-59	An error occurred while trying to rename the file.
-61	Write permissions error.
-108	Insufficient amount of memory.
-120	Directory not found.
-23000	Unable to initialize the local network handler.
-23001	The manually set address is configured improperly.
-23002	A configuration resource is missing.
-23003	Not enough room in the application heap to load MacTCP.
-23004	Error in getting an address from a server or the address is already in use by another machine.

- 23005 A TCPClose command was already issued so there is no more data to send on this connection.
- 23006 The total amount of data described by the WDS was either 0 or greater than 65,535 bytes.
- 23007 The TCP or UDP stream already has an open connection.
- 23008 This TCP stream has no open connection.
- 23009 Maximum TCP or UDP streams are already open.
- 23010 The specified TCP or UDP stream is not open.
- 23011 An open stream is already using this receive buffer area.
- 23012 The TCP connection was broken.
- 23013 The receive buffer area pointer is 0.
- 23014 Invalid RDS or WDS buffers.
- 23015 The connection came halfway up and then failed.
- 23016 The specified command action was not completed in the specified time period.
- 23017 A stream is already open using this local UDP port or a TCP connection already exists between this local IP address and TCP port, and the specified remote IP address and TCP port.
- 23032 The packet is too large to send without fragmenting and the Don't Fragment flag is set.
- 23033 The destination host is not responding to address resolution requests.
- 23035 The icmp echo packet was not responded to in the indicated timeout period.
- 23036 Insufficient internal driver buffers available to fragment this packet on send.
- 23037 No gateway available to manage routing of packets to off-network destinations.
- 23041 The hostName field had a syntax error. The address was given in dot notation (that is, W.X.Y.Z) and did not conform to the syntax for an IP address.
- 23042 The name specified cannot be found in the cache. The domain name resolver will now query the domain name server and return the answer in the call-back procedure.
- 23043 No result procedure is passed to the address translation call when the resolver must be used to find the address.
- 23044 No name server can be found for the specified name string.
- 23045 This domain name does not exist.
- 23046 None of the known name servers are responding.
- 23047 The domain name server has returned an error.
- 23048 Not enough memory is available to issue the needed DNR query or to build the DNR cache.

### **Codes d'erreurs WinSock**

- 10004 Blocking call cancelled
- 10013 Permission denied
- 10014 Bad address
- 10022 Invalid argument
- 10024 No more sockets available
- 10035 Non-blocking socket would block
- 10036 Illegal WinSock function invoked while a blocking function is in progress
- 10037 An attempt was made to cancel an asynchronous operation that has already completed

- 10038 Specified socket descriptor is not valid for this application
- 10039 Destination address was required but none was supplied to the function
- 10040 Datagram too large for buffer
- 10041 Specified protocol does not match the other parameters in the call
- 10042 Protocol option is unknown or invalid
- 10043 Specified protocol is not supported by the Windows Sockets implementation
- 10044 Specified socket type is not supported by the specified address family
- 10045 Socket does not support the specified operation
- 10046 Protocol family not supported
- 10047 Specified address family is not supported by the Windows Sockets implementation or cannot be used with the indicated socket
- 10048 Specified address is already in use
- 10049 Specified address is not available from the local machine
- 10050 Problem with the network subsystem
- 10051 Network cannot be reached from this host at this time
- 10052 Connection was dropped and must be reset
- 10053 Connection was aborted because of a timeout or other error condition
- 10054 Connection was reset by the remote host
- 10055 Windows Sockets implementation is out of buffer space or the space provided in an API call by the application was too small to hold the requested information
- 10056 Specified socket is already connected
- 10057 Specified socket is not connected
- 10058 Socket has had the requested functionality shut down
- 10060 Connection attempt timed out before the connection could be established
- 10061 Connection attempt was forcefully rejected
- 10091 Network subsystem is not yet ready for communication
- 10092 Windows Sockets DLL does not support the requested Winsock protocol version
- 10093 Windows Sockets not initialized
- 11001 Requested database information does not exist; as confirmed by an authoritative host
- 11002 Requested information was not found but the answer was not authoritative
- 11003 Non-recoverable error occurred
- 11004 Name supplied was valid but no information of the requested type is in the database

### Codification SMTP

Les valeurs suivantes **ne sont pas** des codes d'erreurs retournés par les commandes Internet de 4D. Ce sont les codes de réponses définis pour le protocole SMTP afin d'indiquer l'état courant de la communication en mode client-serveur. Cette liste est particulièrement utile dans le cas où vous souhaitez créer votre propre système de messagerie à l'aide des commandes TCP de bas niveau.

- 211 System status, or system help reply
- 214 Help message [Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user]
- 220 <domain> Service ready
- 221 <domain> Service closing transmission channel

250 Requested mail action okay, completed  
 251 User not local; will forward to <forward-path>  
 354 Start mail input; end with <CRLF>.<CRLF>  
 421 <domain> Service not available, closing transmission channel [This may be a  
 reply to any command if the service knows it must shut down]  
 450 Requested mail action not taken: mailbox unavailable [E.g., mailbox busy]  
 451 Requested action aborted: local error in processing  
 452 Requested action not taken: insufficient system storage  
 500 Syntax error, command unrecognized [This may include errors such as command  
 line too long]  
 501 Syntax error in parameters or arguments  
 502 Command not implemented  
 503 Bad sequence of commands  
 504 Command parameter not implemented  
 550 Requested action not taken: mailbox unavailable [E.g., mailbox not found, no  
 access]  
 551 User not local; please try <forward-path>  
 552 Requested mail action aborted: exceeded storage allocation  
 553 Requested action not taken: mailbox name not allowed [E.g., mailbox syntax  
 incorrect]  
 554 Transaction failed

### Codification FTP

Les valeurs suivantes **ne sont pas** des codes d'erreurs retournés par les commandes Internet de 4D. Ce sont les codes de réponses définis pour le protocole FTP afin d'indiquer l'état courant de la communication en mode client-serveur. Cette liste est particulièrement utile dans le cas où vous souhaitez créer votre propre système de transfert de fichiers à l'aide des commandes TCP de bas niveau.

110 Restart marker reply. In this case, the text is exact and not left to the particular  
 implementation; it must read:  
 MARK yyyy = mmmm  
 Where yyyy is User-process data stream marker, and mmmm server's equivalent  
 marker (note the spaces between markers and "=").  
 120 Service ready in nnn minutes.  
 125 Data connection already open; transfer starting.  
 150 File status okay; about to open data connection.  
 200 Command okay.  
 202 Command not implemented, superfluous at this site.  
 211 System status, or system help reply.  
 212 Directory status.  
 213 File status.  
 214 Help message on how to use the server or the meaning of a particular non-  
 standard command. This reply is useful only to the human user.  
 215 NAME system type. Where NAME is an official system name from the list in the  
 Assigned Numbers document.  
 220 Service ready for new user.

221 Service closing control connection. Logged out if appropriate.  
 225 Data connection open; no transfer in progress.  
 226 Closing data connection. Requested file action successful (for example, file transfer or file abort).  
 227 Entering Passive Mode (h1,h2,h3,h4,p1,p2).  
 230 User logged in, proceed.  
 250 Requested file action okay, completed.  
 257 "PATHNAME" created.  
 331 User name okay, need password.  
 332 Need account for login.  
 350 Requested file action pending further information.  
 421 Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.  
 425 Can't open data connection.  
 426 Connection closed; transfer aborted.  
 450 Requested file action not taken. File unavailable (file busy).  
 451 Requested action aborted: local error in processing.  
 452 Requested action not taken. Insufficient storage space in system.  
 500 Syntax error, command unrecognized. This may include errors such as command line too long.  
 501 Syntax error in parameters or arguments.  
 502 Command not implemented.  
 503 Bad sequence of commands.  
 504 Command not implemented for that parameter.  
 530 Not logged in.  
 532 Need account for storing files.  
 550 Requested action not taken. File unavailable (e.g., file not found, no access).  
 551 Requested action aborted: page type unknown.  
 552 Requested file action aborted. Exceeded storage allocation (for current directory or dataset).  
 553 Requested action not taken. File name not allowed.

Les références suivantes sont les URLs complémentaires relatifs aux protocoles Internet. Les pages Web sont accessibles à l'aide de logiciels de navigation tels que Netscape, Internet Explorer ou Mosaic.

**<http://rs.internic.net/>**

Pour comprendre ce qu'est un nom de domaine et connaître la démarche pour en enregistrer un.

**<http://www.ietf.org/>**

Site de l'Internet Engineering Task Force (IETF).

**<http://www.rfc-editor.org/>**

Pour comprendre ce qu'est une RFC et savoir comment les rechercher (voir aussi

**<http://www.rfc-editor.org/rfc.html>**).

**<ftp://ftp.isi.edu/in-notes/rfc821.txt>**

Simple Mail Transfer Protocol -- RFC 821.

**<http://www.w3c.org/>**

Tout ce qu'il faut savoir sur le World Wide Web.

**<http://www.con.wesleyan.edu/~triemer/network/docservs.html>**

Liste de tous les numéros de port TCP/IP avec des liens sur des sources d'informations supplémentaires concernant de nombreux protocoles. Si vous devez travailler avec des routines TCP de bas niveau, c'est le site à connaître.

# Index des commandes

## F

FTP_Append.....	120
FTP_ChangeDir.....	108
FTP_Delete.....	115
FTP_GetDirList.....	106
FTP_GetFileInfo.....	110
FTP_GetPassive.....	128
FTP_GetType.....	121
FTP_Login.....	104
FTP_Logout.....	126
FTP_MacBinary.....	116
FTP_MakeDir.....	112
FTP_PrintDir.....	109
FTP_Progress.....	102
FTP_Receive.....	124
FTP_RemoveDir.....	113
FTP_Rename.....	114
FTP_Send.....	118
FTP_SetPassive.....	127
FTP_SetType.....	122
FTP_System.....	123
FTP_VerifyID.....	111

## I

IT_Decode.....	168
IT_Encode.....	166
IT_ErrorText.....	165
IT_GetPort.....	173
IT_GetProxy.....	170
IT_GetTimeOut.....	164
IT_MacTCPInit.....	156
IT_MacTCPVer.....	160
IT_MyTCPAddr.....	162
IT_Platform.....	157
IT_SetPort.....	174
IT_SetProxy.....	171

IT_SetTimeOut.....	163
IT_TCPVersion.....	159
IT_Version.....	158

## M

MSG_Delete.....	97
MSG_Extract.....	96
MSG_FindHeader.....	90
MSG_GetBody.....	93
MSG_GetHeaders.....	92
MSG_GetMessage.....	94
MSG_HasAttach.....	95
MSG_MessageSize.....	91

## N

NET_AddrToName.....	150
NET_Finger.....	144
NET_NameToAddr.....	149
NET_Ping.....	146
NET_Resolve.....	151
NET_Time.....	147

## P

POP3_BoxInfo.....	75
POP3_Charset.....	85
POP3_Delete.....	73
POP3_Download.....	82
POP3_GetMessage.....	77
POP3_GetPrefs.....	68
POP3_Login.....	69
POP3_Logout.....	74
POP3_MsgInfo.....	76
POP3_MsgLst.....	80
POP3_MsgLstInfo.....	78



POP3_Reset.....	72
POP3_SetPrefs.....	67
POP3_UIDToNum.....	84
POP3_VerifyID.....	71

## S

SMTP_AddHeader.....	52
SMTP_Attachment.....	59
SMTP_Bcc.....	44
SMTP_Body.....	56
SMTP_Charset.....	61
SMTP_Clear.....	32
SMTP_Comments.....	48
SMTP_Date.....	33
SMTP_Encrypted.....	51
SMTP_From.....	36
SMTP_GetPrefs.....	26
SMTP_Host.....	30
SMTP_InReplyTo.....	46
SMTP_Keywords.....	50
SMTP_New.....	29
SMTP_QuickSend.....	27
SMTP_References.....	47
SMTP_ReplyTo.....	40
SMTP_Send.....	31
SMTP_Sender.....	38
SMTP_SetPrefs.....	23
SMTP_Subject.....	54

## T

TCP_Close.....	140
TCP_Listen.....	133
TCP_Open.....	132
TCP_Receive.....	136
TCP_ReceiveBLOB.....	138
TCP_Send.....	135

TCP_SendBLOB.....	137
TCP_State.....	139