XL DOS

EXTENDED BASIC DOS

par D.Sebbag et F.Broche

GUIDE DE L'UTILISATEUR

Félicitations! Vous êtes l'heureux possesseur du XL DOS. Ce dernier va enfin réellement exploiter à fond toutes les possibilités supplémentaires des disquettes par rapport aux cassettes et donner une raison d'exister à votre MICRODISC ORIC. Pour cela, des pages écran vous permettront d'utiliser les nombreuses et puissantes commandes du DOS. Elles sont votre guide, la feuille accompagnant la disquette ne servant qu'à la mise en route. Voyons d'abord comment les utiliser...

Chapitre I

LES PAGES ÉCRAN

Tout d'abord, deux remarques très importantes, puisqu'elles concernent aussi bien les pages guide que les commandes de l'XL DOS:
-Pour l'XL DOS, les minuscules sont équivalentes aux majuscules.
-Tout mot non reconnu comme mot-clé BASIC est considéré comme une commande de l'XL DOS.

Ceci a comme conséquence immédiate que l'emploi du '!', même s'il reste possible, est inutile; les commandes de l'XL DOS font partie, du moins vis à vis de l'utilisateur, des commandes classiques du BASIC et ne font plus surajoutées, comme c'est le cas sur le DOS V1.1, ainsi que sur tous les logiciels ayant vocation de langages et tournant sur Oric... Le '!' reste obligatoire après un THEN et un ELSE.

Les pages écran:

L'XL DOS est composé d'une cinquantaine de commandes. En dehors de ces 50 mots-clés, tout mot frappé au clavier, est considéré comme un fichier, qui sera chargé et exécuté ou non (voir pages Chargement direct).

D'autre part, les pages écran sont des fichiers, des programmes si vous préférez (ces notions ne vous sont de toute façon pas étrangères, puisqu'elles sont valables avec tous les DOS, dont celui livré d'origine avec votre lecteur). Pour voir une page écran, tapez simplement au clavier son nom (nom de fichier).
Exemple: Vous voulez des explications sur les messages d'erreurs générés par l'XL DOS, tapez simplement: ERREUR puis 'Return' ou...
erreur puis 'Return'.

La liste des pages écran vous est donnée en fin de guide, ainsi qu'une description sommaire du domaine traité.

Pour avoir des renseignements sur une commande précise du DOS, la démarche est similaire, à une nuance près: Si vous tapez au clavier une commande de l'XL DOS, elle sera exécutée comme telle. Les noms des pages écran concernant les commandes de l'XL DOS sont alors précédées d'un point, ce qui permet de les distinguer (vous verrez plus tard que le point sert en général à distinguer une commande de l'XL DOS d'un nom de fichier).

Par exemple, si vous désirez des renseignements sur la commande LOAD, tapez:
.LOAD puis 'Return'.
Lorsque le nom de la commande comporte plus de 6 caractères, il ne faut taper que les 6 premiers. La liste des commandes du DOS est donnée dans le chapitre 3.

Comment 'feuilleter' le guide:
Vous avez remarqué la zone inférieure de l'écran: Elle indique les touches à presser pour les différentes options qui vous sont offertes:

'ESC' pour sortir et retourner au BASIC
'Return' pour se replacer au début du guide
Les touches '<' et '>':

-Dans le guide, elles permettent de passer à la page suivante ou précédente.
-Dans les pages écran, elles permettent de visualiser les pages qui sont signalées dans la zone inférieure; en effet, toutes les pages écran sont chaînées: Une page comporte une page précédente et une page suivante, qui traitent souvent d'un domaine voisin. Cela permet d'avoir une vue d'ensemble et rend ce guide encore plus pratique qu'un manuel puisque l'accès à une explication est immédiat. La zone inférieure de l'écran se présentera dans ce cas sous cette forme:

'<' ou '>' pour LOAD ou UPDATE

la page visualisée étant celle du SAVE dans cet exemple précis.

A noter que comme dans les pages GUIDE, la pression de la touche 'Return' dans une page écran envoie sur la page de présentation du GUIDE.

Maintenant, essayez quelques pages et habituez-vous à les feuilleter. Vous ne risquez rien, surtout que seule la zone écran est utilisée, ce qui fait que vous pouvez consulter le GUIDE même pendant que vous programmez. Votre programme ne sera en aucun cas altéré.

Remarque: Les pages du GUIDE sont numérotées. Pour pouvoir y accéder immédiatement, tapez GUIDE suivi du numéro de la page désirée.

Exemple: Pour accéder directement à la page que vous visualisez actuellement, tapez:
GUIDE9
Lorsque le numéro de page a deux chiffres, tapez GUID suivi de ce numéro.

Exemple: Pour la page 12, tapez:
GUID12

Chapitre II

DESCRIPTION GÉNÉRALE

L'XL DOS qui se trouve sur votre disquette se compose en fait d'un programme qui se place dans la RAM overlay (les 16 K libérés par le contrôleur de votre MICRODISC) et de fichiers esclaves, l'ensemble formant les différentes commandes de l'XL DOS. Cependant, les informations contenues sur la disquette sont protégées, même si elles sont toutes accessibles par l'XL DOS. La plupart des commandes d'utilisation courante se trouvent en mémoire. On y trouve aussi des routines spéciales, qui ne sont pas des commandes, mais sont utilisées régulièrement sans que vous ne vous en rendiez compte. Ces routines très pratiques modifient votre contexte de programmation et interviennent plus ou moins profondément dans l'interpréteur BASIC et le système d'exploitation de l'Oric.

Citons par exemple la routine qui permet de considérer les commandes de l'XL DOS comme des mots-clés (pas de '!'), mais aussi la génération automatique des numéros de ligne, la gestion du BRK (pour les programmeurs en langage machine), les touches de fonctions sur Atmos etc...

Chapitre III

LES COMMANDES

Les commandes de l'XL DOS sont de trois types principaux:
-Les commandes de travail sur disque.
-Les commandes d'aides à la programmation et du BASIC étendu.
-Les commandes de gestion de fichiers.

Voyons maintenant en détail le type des différentes commandes.

1) Les commandes de travail sur disque:

LOAD :Chargement de fichiers
SAVE :Sauvegarde de fichiers
UPDATE :Mise à jour de fichiers
DIR :Catalogue de fichiers
DEL :Effacement de fichiers
REN :Changement de nom
PROT :Protection de fichiers
FORMAT :Formatage de disquette
COPY :Copie de fichiers
BACKUP :Copie de disquette(s)
SYS :Configuration de disquette
INIT :Initialisation de disquette
DRV :Disquette par défaut
EXTNS :Extension par défaut

2) Les commandes d'aide à la programmation et du BASIC étendu:

Aide à la programmation:

OLD :Récupération de programmes
RENUM :Renumérotation de lignes
MERGE :Fusion de fichiers
DELETE :Effacement de lignes
NUM :Numérotation automatique

BASIC étendu:

ACCEPT :Saisie de texte formatée
RESTORE :Positionnement DATA
SWAP :Echange de variables

CODE :Codage BASIC
EXECUTE :Exécution de chaîne BASIC
ANGLE :Instruction graphique de type LOGO
ROT :Instruction graphique de type LOGO
BOX :Instruction graphique de type LOGO

Commandes système:

ON :Gestion du '!'
OFF :Gestion du '!'
FUNC :Touches de fonction
PRINTER :Périphérique imprimante
RESET :Boot de l'XT DOS
SEI :Autorisation des interruptions
CLI :Autorisation des interruptions

3) Commandes de gestion de fichiers

FILE :Fichier par défaut

Fichiers 'matrice'

MSAVE :Sauver des tableaux
MLOAD :Rappel des tableaux

Fichiers 'séquentiels'

CLOSE :Fermeture de fichiers
FSTART :Retour début de fichier
FJUMP :Saut d'enregistrements
FEND :Saut à la fin du fichier
PUT :Ecriture sur le fichier
TAKE :Lecture de variables
OPEN"L" :Ouverture de fichiers

Fichier à accès direct

OPEN"R" :Ouverture du fichier
CLOSE :Fermeture du fichier
FIELD :Définition des champs
LSET :Ecriture dans les champs
RSET :Ecriture dans les champs
PUT :Ecriture d'une fiche
TAKE :Lecture d'une fiche
&( ) :Nombre de fiches

Fichier 'chaîne' et 'disque'

OPEN"S" :Ouverture du pseudo fichier
OPEN"D" :Ouverture du fichier disque

ANNEXE: LISTE DES PAGES ECRAN

NOMS :Noms de fichiers
ERREUR :Messages d'erreur
RWTS :Gestion interne disquettes
RAMROM :RAM overlay
FICH :Généralités sur les fichiers
SEQ :Fichiers séquentiels
DIRECT :Fichiers directs
DISQUE :Fichiers disques

CHAINE :Fichiers chaînes
'ESC' :Touche spéciale

NOMS DE FICHIERS

Les noms de fichiers que vous précisez dans les commandes de l'XL DOS ont une structure bien précise. Ils doivent être entourés de guillemets, comme des chaînes alphanumériques. Ils comportent en général deux parties:

Le nom proprement dit, qui se compose au plus de 6 caractères alphanumériques (symboles exclus) mais peut n'en comporter aucun puisque chaque caractère absent est considéré par l'XT DOS comme un espace.

L'extension, qui a au plus 3 caractères, le principe restant similaire à celui de la partie nom.

Chacune de ces parties est optionnelle.
Lorsqu'on désire que la commande ait un effet sur un lecteur distinct du lecteur par défaut, le numéro du lecteur souhaité doit être précisé juste après le guillemet et suivi d'un tiret. Les numéros de lecteur vont de 0 à 3.

Exemples:
"2-TOTO.COM" est valide et la commande agira, à condition que le lecteur 2 soit présent.
"PROGRAMME.BAS" est invalide, car la partie nom comporte plus de 6 caractères.
"TOTO-.BIN" est invalide, car le nom comporte un caractère non alphanumérique.

Les caractères 'jokers':
Dans certaines commandes, il peut être utile de préciser plusieurs noms de fichiers à la fois. **Par exemple** dans un COPY, lorsque vous souhaitez copier tous les fichiers ayant une extension .COM. Deux caractères peuvent être utilisés à cet effet, c'est ce que l'on appelle les caractères 'jokers'. Il s'agit de '*' et '?'. Le caractère '*' désigne un groupe de caractères et doit être placé à la fin du nom ou de l'extension. Le caractère '?' désigne un unique caractère quelconque.

Exemples: Vous souhaitez désigner tous les noms de fichiers commençant par AB et ayant une extension quelconque: Le nom sera: "AB*.*"
Vous voulez tous les noms de deux caractères et dont le second sera un Q: Le nom sera "?Q"

Remarque: Les noms de fichiers peuvent être des variables alphanumériques: A$, avec A$="TOTO.COM" est un nom de fichier valide.

Erreurs: 'Invalid filename error' lorsqu'un nom de fichier invalide a été spécifié.

GESTION DES ERREURS

Le DOS possède une trentaine de messages d'erreur. Il sont en minuscule.
Tous ces nouveaux messages peuvent être 'interceptés', de manière qu'ils n'interrompent pas le déroulement du programme: Il suffit de faire:

POKE#4FD,1
POKE#4FD,0 les autorise à nouveau.

Dans tous les cas, le numéro de l'erreur se trouve en #4FF.
S'il s'agit de FDERR, le code de l'erreur d'entrée/sortie se trouve en #4FE.

Voici la liste des messages:
01 File not found
02 Invalid command end
03 No drive number
04 Bad drive number
05 Invalid file name
06 FDERR= (code d'erreur)

07 Illégal attribute
08 Wildcard(s) not allowed
09 File already exists
10 Insufficient disc space
11 File open
12 Illegal quantity
13 End adress missing
14 Start adress > end adress
15 ?
16 Renamed file not on same disc
17 Unknown array
18 Target drive not source drive
19 Destination not specified
20 Cannot merge and overwrite
21 Single target file illegal
22 Syntax
23 File name missing
24 Source file missing
25 ?
26 Disc write-protected
27 Incompatible drives
28 File not open
29 File end
30 Bad file mode
31 Bad record number
32 Field overflow
33 String too long
34 File type
35 Can't open
36 Unknown field name

Certaines erreurs particulières seront traitées avec les instructions qui risquent de les provoquer. D'autres peuvent se produire pour toutes les commandes, elles seront expliquées une fois pour toutes ici.

'File not found' :Le fichier demandé n'est pas sur la disquette.
'Invalid command end' :L'option précisée est invalide.
'No drive number' :Tentative d'opération sur un drive non connecté.
'FDERR=' :Erreur système: Disquette ou connections défectueuses.
'Wildcard(s) not allowed' :Caractères 'jokers' interdits.
'Insufficient disc space' :Tous les secteurs de la disquette sont occupés.
'Illegal quantity' :Paramètre spécifié trop grand.
'Syntax' :Rien à signaler...
'File name missing' :L'XL DOS voulait un nom de fichier.
'Type mismatch' :Une chaîne à la place d'un nombre et vice versa.
'Disc write protected' :Tentative d'écriture sur un disque protégé par la languette.
'Illegal attribute' :Option de protection invalide.

RWTS

RWTS veut dire: Read Write Track Sector. C'est à dire lecture ou écriture des pistes/secteurs. C'est le nom donné aux routines du DOS chargées des opérations de contrôle des disquettes. Pour ce faire, elles agissent sur les registres du contrôleur de disque:

#310 :Registre de commande drive
#311 :Registre de piste
#312 :Registre de secteur
#313 :Registre de données
#314 :Commande interface & contrôleur
#315-317 :Commande interface & contrôleur
#318 :Donnée prête

#319-3FF :Image VIA ou extensions...

RAM OVERLAY

En plus de 48 Ko normalement disponibles, l'Oric possède 16 Ko supplémentaires de RAM, placés aux mêmes adresses que la ROM (#C000-#FFFF). C'est ce que l'on nomme la RAM en OVERLAY.

Le lecteur de disquette utilise la RAM overlay pour y mettre le DOS. Mais pour pouvoir utiliser la RAM et la ROM, il faut des routines qui assurent le passage de l'une sur l'autre. Ces routines sont implantées dans la zone #400-#4FF. Il convient donc de n'utiliser cette zone sous aucun prétexte: l'Oric ne s'y retrouverait pas! Toutefois, l'analyse de ces routines vous aidera à comprendre le fonctionnement du passage RAM/ROM.

GESTION DE FICHIERS

Vous l'aurez constaté, le mot fichier est employé très souvent dans ce manuel. En fait, le mot fichier revêt deux significations différentes:
-Sens général: Est appelé fichier tout ce qui a un nom sur le catalogue de la disquette.
-Sens particulier: Suite de données ayant une signification pratique: Fichier adresses, Fichiers clients...

On donne le nom de 'gestion de fichiers' à l'élaboration de fichiers de la seconde catégorie.

Les fichiers dont nous allons parler sont donc un ensemble de données enregistrées sur disque, organisées de manière à pouvoir être relues en mémoire centrale plus ou moins facilement.
Cas le plus courant de fichier: Un fichier adresses, qui comportera nom, adresse et numéro de téléphone.

Les fichiers sont principalement de deux types classiques:
-Accès séquentiel: Le plus simple à mettre en oeuvre, c'est aussi le moins performant: Toutes les données sont stockées les unes après les autres, on accède à une donnée après avoir lu toutes les précédentes: Facilité d'utilisation, mais lenteur intrinsèque.
-Accès direct: Les variables ne sont pas organisées de manière séquentielle, mais en fiches, auxquelles on peut accéder directement.

L'XL DOS autorise deux autres types de fichiers, que l'on trouve rarement même sur de gros systèmes:
-Les fichiers chaînes: Un peu semblables, dans la forme, aux fichiers à accès direct, ce sont des fichiers qui ne font pas appel aux disques permettant de faciliter le travail sur les chaînes de caractères.
-Les fichiers disques: Eux aussi assez proches des fichiers directs. Ils permettent de travailler directement sur les secteurs de la disquette: A réserver aux utilisateurs avertis !!!

ORGANISATION GENERALE

L'XL DOS utilise un système original pour gérer les fichiers, qui lui permet de n'utiliser que la mémoire strictement disponible. Il stocke les informations nécessaires dans l'espace utilisateur, de manière tout à fait transparente pour le BASIC.

Attention: Un clear efface toutes les données des fichiers et rend donc impossible la fermeture des fichiers. Cet inconvénient n'en est un que lors de la mise au point des programmes, la méthode de travail étant en revanche plus performante: Problème de choix donc (N'oubliez pas que toute modification de ligne BASIC effectue un CLEAR par exemple.)

Toute la gestion de fichiers s'organise en fait autour de tampons de longueur fixe (254 pour le séquentiel, longueur d'un secteur pour l'accès disque) ou variable pour les fichiers directs ou chaînes.

A chaque tampon est affecté un numéro, appelé numéro logique du fichier (en abrégé NL).
Lors de l'ouverture d'un fichier direct ou séquentiel, vous affecterez donc un nom de fichier (6 caractères dans ce cas) à un numéro logique. Un numéro logique peut prendre des valeurs de 1 à 254. Le fichier numéro 0 est en fait réservé par le système pour la sauvegarde des matrices (Cf. MSAVE, MLOAD).

Procédure générale d'opération sur les fichiers:
1-Il faut, grâce à l'ordre OPEN, 'ouvrir' le fichier, de manière à ce que l'Oric prenne certains repères sur le disque notamment et les appelle en mémoire centrale. Bien entendu, un numéro logique ne peut être affecté qu'à un seul fichier...
2-Vous pouvez maintenant lire ou écrire sur le fichier (pour les instructions de lecture/écriture, voir les rubriques SEQ, DIRECT, DISQUE et CHAINE).
3-En fin de session, il faut 'fermer' le fichier (ordre 'CLOSE') pour libérer les tampons et écrire sur le disque les informations appelées en mémoire centrale par 'OPEN'.

La place manquant pour faire un véritable cours sur les fichiers, nous invitons les utilisateurs à se reporter à l'excellent livre de J. BOISGONTIER, paru aux éditions P.S.I.: 'Le BASIC et ses fichiers'.

Erreurs: 'File open' :Tentative d'ouverture d'un fichier déjà ouvert.
'File not open' :Tentative d'opération sur un fichier non encore ouvert.
'Bad file mode' :Opération incompatible avec le type de fichier
(LSET pour un fichier séquentiel par exemple).

SEQUENTIEL

Un fichier séquentiel est un fichier dans lequel on relit les informations dans l'ordre où elles ont été stockées. Ces informations sont des variables de tous types (réel, entier, chaîne).

Pour un peu faciliter les choses, vous gérez en fait un pointeur de fichier, qui vous dit où vous lisez ou écrivez. Lorsque vous ouvrez un fichier, le pointeur est placé au début du fichier.

Trois commandes agissent directement sur la position du pointeur:
FSTART :Place le pointeur en début de fichier.
FEND :Place le pointeur en fin de fichier.
FJUMP :Déplace le pointeur d'un certain nombre d'enregistrements.

D'autres commandes agissent indirectement sur le pointeur de fichier, ce sont l'écriture et la lecture, qui déplacent le pointeur sur l'enregistrement suivant.

MANIPULATION D'UN FICHIER SEQUENTIEL

Comme d'habitude, il convient d'ouvrir le fichier (voir OPEN). Pour que l'ouverture soit acceptée, il faut que le fichier ne soit pas ouvert déjà (impossible d'ouvrir un même fichier séquentiel sous plusieurs numéros logiques).

La lecture de la variable courante se fait à la position courante du pointeur par TAKE.
Exemple: TAKE #1,A,B$.
Si la fin du fichier est atteinte, le message 'END of file' est généré. Pour éviter ce genre de désagrément, il suffit de tester la variable logique &(NM) qui est vraie si la fin du fichier est atteinte. La variable &(NL) contient le type de la variable courante: 255= réel, sinon la longueur de la chaîne alphanumérique.

L'écriture se fait à la position courante du pointeur. Deux cas de figure:
-Si l'écriture se fait à la fin du fichier, dans la zone ou rien n'a été écrit, tous types de variables peuvent être écrites.
-Si l'écriture se fait au milieu du fichier, écrasant des données déjà existantes, les variables écrites et écrasées doivent être de même type, car la place correspondante a déjà été allouée. Dans le cas contraire, un 'TYPE MISMATCH ERROR' est généré.

Dans le cas de chaînes, la nouvelle écrite sera tronquée à droite si elle est trop longue ou justifiée à droite par des espaces si elle est trop courte.

DIRECT

Dans ce type de fichier, les variables ne sont pas organisées de manière séquentielle, mais en fiches, elles mêmes subdivisées en zones.

Une notion fondamentale est celle de tampon: Les lectures/écritures se feront par l'intermédiaire d'un tampon de

la longueur d'une fiche.

Vous disposez donc de deux types d'instructions:
I Sauvegarde/Rappel d'une fiche.
II Travail sur le tampon.

Voyons en détail ces deux types d'instructions:

I Sauvegarde/Rappel d'une fiche.

Le transfert Disque>Tampon se fait par TAKE (#NL,)Numéro de fiche
Le transfert Tampon>Disque se fait par PUT (#NL,)Numéro de fiche

Le numéro de fiche doit être au plus égal au numéro de la dernière fiche +1. Le nombre de fiches ne peut excéder 65535.

Le numéro de la dernière fiche est donné par &(NL). La longueur d'une fiche par &(-NL).

Attention: Pour copier un fichier, il ne faut copier que le fichier d'extension .DAT.

II Travail sur le tampon.

Le tampon dans lequel vous écrivez a la longueur de la fiche, c'est aussi la longueur donnée lors de l'ouverture du fichier.

A l'intérieur de ce tampon, on définit des CHAMPS. Ce sont des zones de la fiche. On distingue trois types de champs: Entiers, réels et alphanumérique. Ces champs ont respectivement pour longueur effective: 2, 5, longueur de la chaîne. Il faut ajouter à ces longueurs un octet séparateur par champ. Ne pas oublier d'en tenir compte lors de la déclaration de la longueur d'une fiche.

Chaque type de champ a un code:
0 :Pour un champ entier,
255 :Pour un champ réel,
1 à 254 :Pour un champ alphanumérique (longueur).

Un nom de champ peut avoir au maximum 6 caractères significatifs et peut être un tableau à une dimension dont le nombre d'éléments est limité à 256.
Exemple de nom de champ: NOM, PRENOM, CADRE(I) etc...
NB: CADRE(0) <=> CADRE

Attention: Malgré leur apparence formellement identique, il ne faut pas confondre CHAMP et variable, nous allons voir comment on passe de l'un à l'autre: Pour écrire des données dans un champ, il faut utiliser les instructions LSET et RSET. Totalement équivalentes pour les champs numériques, ces 2 instructions agissent différemment sur les champs alphanumériques: Elles justifient à droite, respectivement à gauche les chaînes par des espaces ou les tronquent à droite, respectivement à gauche si elles sont trop longues. La définition des champs se fait avec l'instruction FIELD. La lecture d'un champ, autrement dit le transfert de sa valeur dans une variable est effectué par l'opérateur '>'.

Exemples:

OPEN "R","ESSAI",100,1
Ouvre le fichier "ESSAI" sous le numéro logique 1, avec une longueur de 100.

FIELD #1,NOM TO 14
Définit un champ nom, au début de la fiche, de 14 caractères.

LSET NOM < "DUPONT Jean"
Ecrit dans le tampon une variable.

NOM > E$
Transfère dans la variable E$ la valeur du tampon, soit ici: ?E$

"DUPONT Jean "

RSET au lieu de LSET aurait donné E$=" DUPONT Jean"

DISQUE

Voici un type très particulier de fichier. Il permet à ceux qui connaissent la structure d'une disquette de travailler directement dessus à partir du BASIC.

Le principe est simple: On ouvre une sorte de fichier direct dont la longueur est celle d'un secteur.

Seules diffèrent l'écriture et la lecture du tampon sur disque: Au lieu de préciser des numéros de fiches, on précisera directement l'adresse piste secteur du secteur à écrire. Pour modifier le tampon, vous disposez des instructions: FIELD, <, LSET, RSET.

Détail des instructions de lecture/écriture:

TAKE (#NL,)Piste,Secteur
Rien à signaler...

PUT (#NL,)Piste,Secteur,Octet0,Octet1,Longueur secteur
Structure d'un secteur: Octet0, Octet1, Longueur effective du secteur.

Vous lisez/écrivez en fait les 254 derniers octets du secteur.

Les 2 premiers sont accessibles en lecture par &(NM) et &(-NL) et en écriture lors du PUT par Octet0 et Octet1. Longueur secteur est le troisième octet du secteur, il a été isolé.

Attention: Les champs ne contiennent pas de séparateur.

CHAÎNE

Ce quatrième et dernier type de fichier permet de simplifier le travail sur les chaînes de caractères.

C'est une sorte de fichier direct qui n'aurait pas de correspondance sur disque:

Même type de travail sur les tampons que pour les fichiers à accès direct ou à accès disque, mais pas d'instructions de lecture/écriture sur la disquette.

Une excellente occasion de se familiariser avec le travail sur les tampons!!!

ESC

La touche 'ESC' a une utilisation particulière dans toutes les commandes de l'XL DOS et dans les commandes annexes (INIT, SYS...).

Elle permet de sortir directement de la commande en cours, ce qui peut être souvent très pratique.

Le rôle particulier de la touche 'ESC' n'est pas rappelé dans toutes les commandes.

Rappel: Lors d'un ACCEPT (voir ce mot-clé), la touche 'ESC' a aussi un rôle particulier. Dans tous les cas, si vous voulez générer des attributs vidéo, 'CTRL/Z' est valide.

LES COMMANDES DE L' XL DOS

LOAD

SYNTAXE: (!)LOAD Nom de fichier,options

Chargement de fichier.

Les options sont:
,J pour chaîner 2 programmes
,D pour visualiser les adresse
,N pour inhiber l'exécution automatique
,A adresse pour charger à partir de l'adresse spécifiée.

Chargement direct:
Il est possible de charger un fichier directement sans passer par la commande LOAD: En effet, tout mot tapé au clavier, s'il n'est pas une commande BASIC ou XL DOS est considéré comme un nom de fichier à charger.

Donc, pour charger par exemple le fichier RIRI.TXT, taper simplement RIRI.TXT (on ne peut pas faire plus simple).

Lorsqu' aucune extension n'est spécifiée, l'extension .COM est prise par défaut. Par exemple, pour charger le fichier ZORGON.COM, taper ZORGON puis 'Return'.

Lorsqu'un nom de fichier commence par un mot-clé du BASIC ou du DOS (exemple ONDE.BAS), il doit être précédé d'un point afin de le distinguer du BASIC. Donc il faudra taper .ONDE.BAS. Ces contraintes ne sont pas applicables à la commande LOAD, qui permet en outre de spécifier les options vues au début.

En mode OFF ('!' obligatoire), les noms de fichiers à charger directement doivent être aussi précédés du point d'exclamation.

A noter que les noms de fichiers de l'XT DOS, contrairement au DOS V1.1, peuvent comporter des mots-clés BASIC, sans que cela provoque un message 'Invalid filename error'.

SAVE

SYNTAXE: (!)SAVE Nom de fichier,options

Sauvegarde de fichier sur disquette

Les options sont:
,A adresse de début ,E adresse de fin, pour un fichier binaire.
,T adresse de lancement d'un programme en langage machine.
,AUTO pour exécution automatique d'un programme BASIC ou lancement à l'adresse de début d'un programme en langage machine.

Erreurs:
'End address missing' lorsque seule l'adresse de début (option A) a été précisée.
'Start address > end adress' lorsque l'adresse de début est supérieure à l'adresse de fin, précisées par les options ,A et ,E.

UPDATE

SYNTAXE: Identique à celle de SAVE

Cette instruction est très utile pour la mise au point de programmes et la sauvegarde des différentes versions au fur et à mesure de leur création.

Le fichier concerné sera d'abord sauvé, même si un fichier de même nom existait déjà (la version précédente). Puis l'XL DOS cherchera dans le catalogue un fichier de même nom et d'extension .BAK (pour backup, c'est à dire version antérieure). Si ce fichier existe, il est effacé. Ensuite, le fichier qui se trouvait avant le UPDATE sur la disquette (si le programme n'est pas mis à jour pour la première fois) voit son extension changée en .BAK: Il devient la version antérieure.

En somme, le UPDATE vous permet de toujours avoir sur la disquette de travail la dernière version et la version antérieure du programme en cours d'élaboration, tout ceci étant effectué en une seule instruction. Sans celle-ci, vous seriez obligé pour le même résultat de taper une suite de DEL, REN, SAVE.

Exemple: Vous travaillez sur le fichier RIRI.COM. Supposons que ce dernier n'existe pas à la première sauvegarde.

UPDATE"RIRI.COM"
Catalogue: RIRI.COM 1ère version

UPDATE"RIRI.COM"
Catalogue: RIRI.BAK 1ère version
RIRI.COM 2ème version

UPDATE"RIRI.COM"
Catalogue: RIRI.COM 3ème version
RIRI.BAK 2ème version
etc...

DIR LDIR

SYNTAXE: (!)DIR (nom de fichier)
(!)LDIR (nom de fichier)

Catalogue de la disquette, sur écran (DIR) ou imprimante (LDIR). Le nombre qui suit le 'Out of' est le nombre de secteurs utilisateur, déduction faite d'un secteur système et des secteurs du DIR. Ce nombre vaut au départ 834, soit 208.5 Koctets par face (augmentation de 49 Koctets par rapport au DOS V1.1).

En effet, l'XL DOS travaille sur 44 pistes de 19 secteurs, au lieu de 19 secteurs, au lieu de 40 pistes de 16 secteurs avec le DOS V1.1.

DEL

SYNTAXE: (!)DEL (nom de fichier)

Détruit le ou les noms de fichiers précisés.

Lorsque plusieurs noms sont précisés grâce aux caractères 'jokers', il est demandé de confirmer la destruction pour chaque fichier.

Répondre 'Y' pour oui.

REN

SYNTAXE: (!)REN Nom fichier TO Nom fichier
Changement de nom de fichier.

(!)REN Nom de fichier (,options)
Reconfiguration d'un fichier:
La syntaxe est pratiquement similaire à celle du SAVE, pour ce qui est des options mais au lieu d'être sauvé, le fichier concerné verra son statut modifié: Adresse de départ, de transfert etc...

Remarque: Les options non précisées laissent le statut dans son état initial.

Exemples:

REN"TOTO.COM",AUTO
Si TOTO.COM est un programme BASIC, il se lancera désormais avec un RUN.
Si TOTO.COM est un fichier machine, il se lancera à l'adresse de début du fichier.

REN"TOTO.COM",Axxxx(,Tyyyy)
TOTO.COM se chargera désormais à partir de xxxx. L'adresse de fin est automatiquement calculée: L'option Exxxx est donc inutile et invalide.

PROT

SYNTAXE: (!)PROT (nom fichier)(,option)

Pas d'option :Protection contre l'écriture (SAVE, DEL...).
Option (,P) :Protection contre le COPY, LOAD avec option...
Option (,I) :Rend le fichier invisible dans le catalogue.
Option (,N) :Annule les options ',I' ou protection en écriture.

Remarque: Les caractères 'jokers' sont autorisés.

Exemples: PROT 1 :Protège toute la disquette 1 contre l'écriture.
PROT "TOTO.COM",P :Protège contre le COPY le fichier TOTO.COM

FORMAT

SYNTAXE: !FORMAT Numéro de disque

Formate la disquette précisée.

Cette opération est indispensable pour toute disquette neuve: Il faut que l'XL DOS puisse se retrouver lors des écritures ou lectures sur la disquette.

Attention: L'opération de formatage est irréversible. Assurez-vous donc que la disquette ne contient aucun fichier intéressant.

Remarque: Les disquettes formatées avec le DOS V1.1 sont compatibles, sauf pour le BACKUP.

COPY

SYNTAXE: (!)COPY (nom fichier)(TO nom fichier)(,options)

Options:
,P et ,N :Donner le statut de protection du fichier qui sera créé.
,C :Lorsque vous ne disposez que d'un lecteur,

permet de changer de disque après chaque opération.
,O :Pour écraser le fichier cible s'il se trouvait déjà sur la disquette.
,M :Pour fusionner plusieurs fichiers en un seul.

L'option ',M' permet de regrouper plusieurs fichiers en un seul, qui sera chargé par conséquence en plusieurs endroits différents de la mémoire. Ceci est très utile avec des programmes qui comportent une partie en BASIC et une partie en langage machine.

Lorsque cette option est spécifiée, les fichiers sources sont accolés au fichier cible. Lorsque cela a été effectué pour tous les fichiers sources (spécifiés au besoin avec des caractères 'jokers'), le message 'Next:' est affiché et l'XL DOS attend que vous lui précisiez un nouveau nom de fichier.

Si vous souhaitez sortir à ce moment, pressez 'Return'. Si vous souhaitez accoler un autre fichier, tapez son nom. Les règles normales d'un nom de fichier sont applicables (spécification d'un numéro de lecteur, suivi d'un tiret etc...).

Erreurs:
'Target drive not source drive' L'option ',C' nécessite l'usage d'un seul drive.
'Destination not specified' L'option ',M' nécessite que soit précisé un fichier cible.
'Single target file illegal' Sauf dans le cas de l'option ',M', il faut que, si le nom du fichier source comporte des 'jokers', le nom du fichier cible soit, s'il est précisé, un numéro de drive.
'Cannot merge and overwrite' Les options Merge ',M' et Overwrite ',O' ne peuvent être utilisées simultanément, ce qui est logique.

BACKUP

SYNTAXE: (!)BACKUP (Numéro drive) (TO Numéro drive)

Recopie la totalité de la disquette et non fichier à fichier comme pour le COPY.

Si Numéro drive n'est pas précisé, le drive courant est pris par défaut (voir aussi DRV).

Si vous utilisez un seul drive, le BACKUP nécessitera 5 manipulations.

Attention: La disquette cible doit être formatée comme la disquette source (44 pistes de 19 secteurs).

SYS

SYNTAXE: (!) SYS

Permet de changer la configuration d'une disquette (son nom, son nombre de pistes etc...).

Cette instruction est en fait un fichier BASIC qui se trouve sur la disquette master et qui sera chargé et exécuté à chaque appel de cette instruction. Il est donc facilement modifiable par l'utilisateur.

INIT

SYNTAXE: (!)INIT (numéro de drive)

Cette commande permet de créer des disquettes 'esclaves', capables d'initialiser le système.

Elle permet aussi de personnaliser une disquette en y écrivant un message de présentation qui sera affiché à chaque 'RESET'.

DRV

SYNTAXE: (!)DRV Numéro de lecteur

Définit le lecteur par défaut, c'est à dire le lecteur qui sera utilisé lorsqu' aucun numéro de drive n'est précisé.

Remarque: Lorsque des instructions qui doivent faire appel à des fichiers esclaves sont utilisées (INIT, COPY etc...), c'est sur le lecteur par défaut que ces fichiers seront recherchés.

EXTNS

SYNTAXE: (!)EXTNS chaîne alphanumérique

Définit l'extension par défaut, c'est à dire, celle qui sera prise en considération lorsqu' elle n'est pas précisée lors d'un chargement direct. La chaîne doit comporter au plus trois caractères de type valide pour les noms de fichiers.

Remarque: L'extension par défaut après une initialisation est .COM (pour 'commande').

OLD

SYNTAXE: (!)OLD

Permet de récupérer un programme perdu à cause d'un NEW ou d'une réinitialisation.
Ceci pour le programme courant (l'adresse est contenue en #9A).

SYNTAXE: (!)OLD adresse

Modifie l'emplacement du programme BASIC.
In '0' est mis à l'adresse spécifiée et les pointeurs BASIC sont ajustés.

Exemple: Après un OLD #5000, #9A vaut #5000 et les autres pointeurs sont ajustés.
Dans tous les cas, OLD effectue un CLEAR (détruire les variables).

RENUM

SYNTAXE: (!)RENUM Pas,Origine,L1,L2

Renumérote un programme, y compris GOTO, GOSUB, THEN, ELSE, RUN.

Paramètres:
Pas :Différence entre deux lignes consécutives, 10 par défaut.
Origine :Nouveau numéro de la première ligne renumérotée, 100 par défaut.
L1 :Numéro de la 1ère ligne à renuméroter, par défaut la 1ère ligne du programme.
L2 :Numéro de la dernière ligne à renuméroter, par défaut la dernière ligne du programme.

Attention: N'appuyer pas sur RESET pendant l'exécution!!

Tous les paramètres peuvent être omis, ils prennent alors leurs valeurs par défaut.

Les SYNTAXES suivantes sont valides: (!)RENUM <=> RENUM 10,100,0,64000
(!)RENUM 20

(!)RENUM 20,500
(!)RENUM ,500
(!)RENUM ,,1000
(!)RENUM ,,,2000
(!)RENUM 20,,40

Si un numéro de ligne est inexistant, le RENUM prendra la ligne suivante.

MERGE

SYNTAXE: (!)MERGE Nom de fichier

Fusionne le programme spécifié avec celui en mémoire, sans perdre les variables.
Les numéros de ligne des programmes peuvent être quelconques.
Dans le cas où les programmes ont des numéros de ligne communs, la ligne initiale sera effacée.
En mode programme, on peut éviter l'affichage des lignes mélangées par POKE #4FD,1.

Erreur: 'File type' si le programme n'est pas un programme BASIC.

DELETE

SYNTAXE: (!)DELETE Ligne1,Ligne2

Détruit sans perte des variables les lignes Ligne1 incluse à Ligne2 non incluse. Si Ligne2<Ligne1, le DOS rétablit l'ordre: DELETE 100,3000 <=> DELETE 3000,100.

SYNTAXE: (!)DELETE Ligne1

Détruit, sans perte des variables, la fin du programme, à partir de la Ligne1 incluse.
SYNTAXE: (!)DELETE ,Ligne2

Détruit sans perte des variables jusqu'à la Ligne2 non comprise.

Erreur: 'UNDEF'D STATEMENT'

NUM

SYNTAXE: (!)NUM
(!)NUM Origine
(!)NUM Origine,Pas
(!)NUM ,Pas

Précise les paramètres initiaux de la renumérotation automatique des lignes BASIC.

Un nouveau numéro est généré par la séquence 'Funct'+'Return'. Cette instruction n'a donc d'intérêt que sur Oric Atmos, l'Oric-1 ne disposant pas de touche de fonction.

ACCEPT

SYNTAXE:
(!)ACCEPT (@X;Y)(,&Code),Variable mode de sortie,Variable chaîne,Longueur(,option)

Cette instruction, d'abord assez complexe, permet de formater à volonté les entrées de texte sur l'écran.

Principe général: Une fenêtre de longueur précisée va être créée à l'écran, à l'intérieur de laquelle tous les déplacements avec les fléches de curseur sont autorisés (éditeur de type pleine page).

Explication des paramètres:

@X;Y :Positionner le début de la fenêtre en absolu, y compris en haute résolution.

,&Code :Préciser le code ASCII de remplissage de la fenêtre.

,Variable mode de sortie
:Selon l'option précisée, vous avez plusieurs modes de sortie de la fenêtre:

-par 'Return' :Variable=0
-par 'ESC' :Variable=1
-par 'curseur G' :Variable=2 (*)
-par 'curseur D' :Variable=3 (*)
-par 'curseur H' :Variable=4 (*)
-par 'curseur B' :Variable=5 (*)

(*) Ces modes de sortie ne sont pas toujours valides, voir options.

,Variable chaîne :Elle sera chargée avec la chaîne entrée (un peu comme pour un INPUT). Sa longueur, quel que soit le nombre de caractère entrés sera celle précisée par la variable longueur. Quel que soit le caractère 'de fond', la chaîne sera justifiée à droite par des espaces.

,Expression longueur :Elle définit la longueur de la fenêtre (1 à 255 caractères).

,Options :Elle peut ne pas être précisée, elle prend alors la valeur '0'.
Cette variable permet de définir:
-Si la fenêtre va être effacée, c'est à dire remplie avec le code ASCII courant (celui-ci n'a pas besoin d'être précisé chaque fois).
-Si on autorise ou non la sortie de la fenêtre par les touches de curseur ou seulement par 'ESC' et 'Return'.

Donc, quatre valeurs possibles:	Effacer	Sortie curseur	Valeur
	oui	oui	0
	non	oui	1
	oui	non	2
	non	non	3

NB: 'ESC' peut être remplacé, pour les attributs vidéo, par 'CTRL/Z'. Ces attributs vidéo, contrairement à un INPUT ne seront pas détruits, mais remplacés par leur code+128: Ils apparaîtront correctement avec PRINT, mais en vidéo inverse lors d'un PLOT.

Exemple: ACCEPT @10;1,&ASC("-"),A,A$,10,2
Crée une fenêtre de 10 caractères, la remplit de '-'. Tapez TOTO puis 'ESC': A vaut 1, A$ vaut "TOTO ". Vous ne pouvez sortir que par 'Return' ou 'ESC'.

RESTORE

SYNTAXE: RESTORE
RESTORE Numéro de ligne

Place le pointeur de DATA au début du programme ou de la ligne précisée. Si une ligne est spécifiée, elle doit exister, mais il n'est pas indispensable qu'elle contienne des DATA.

Erreur: 'UNDEF'D STATEMENT'

SWAP

SYNTAXE: (!)SWAP Variable1,Variable2

Echange les deux variables.

Erreur: TYPE MISMATCH lorsque les variables ne sont pas du même type.

CODE

SYNTAXE: (!)CODE Nom de chaîne

Remplace la chaîne spécifiée par la même chaîne codée (reconnaissance des mots-clés BASIC).
A utiliser avec EXECUTE. Si une même chaîne doit être exécutée plusieurs fois, un seul codage est nécessaire.

Exemple: CODE A$

Erreur: 'String too long' si la chaîne à coder dépasse 77 caractères.

EXECUTE

SYNTAXE: (!)EXECUTE Chaîne

Exécution d'une chaîne, qui doit auparavant avoir été codée par CODE.
Toutes les instructions peuvent être exécutées, y compris test etc...

Exemple: 10 A$="PING":CODE A$
20 EXECUTE A$

Erreurs: 'String too long' si la chaîne dépasse 70 caractères.
'ILLEGAL DIRECT' si tentative d'exécution en mode direct.

ANGLE

SYNTAXE: (!)ANGLE Valeur

Instruction de type LOGO.

Cette instruction, ainsi que les deux qui suivent, permettent de créer des graphiques d'une façon différente d'avec les instructions CURSET et autres.

On donne au curseur un angle initial, grâce à l'instruction ANGLE. Ensuite, on déplace ce curseur d'un certain nombre de points dans la direction précisée, grâce à l'instruction LINE. L'instruction ROT permet de modifier cette direction de façon relative (par rapport à l'angle précédent).

LINE

SYNTAXE: (!)LINE nombre de point,code FB

Déplace le curseur du nombre de points spécifié, dans la direction courante, avec la couleur spécifiée.

Exemple: ANGLE 45:LINE 20,1
Trace une ligne de 20 points dans une direction de 45 degrés à partir de la position courante du curseur.

ROT

SYNTAXE: (!)ROT Angle

Ajoute à la direction courante du curseur LOGO la valeur de l'angle spécifié en degrés. Cette valeur peut être négative.

Exemple:
```
10 HIRES:SEI:ANGLE 0
20 CURSET 120,100,0
30 FOR I=0 TO 240
40 LINE I/2+10,1
50 ROT 91
60 NEXT:LINE 60,1
70 CLI
```

Essayez plusieurs valeurs d'angle à la ligne 50, cela vaut le détour!!!

BOX

SYNTAXE: (!)BOX Largeur,Hauteur,Code FB

Trace un rectangle de largeur et de hauteur spécifiées, dont le sommet haut-gauche est la position courante du curseur.

Exemple:
```
10 HIRES:SEI
20 FOR X=0 TO 5.9 STEP .05
30 CURSET X*33,X*10*SIN(4*X)+99,3
40 BOX 6*X+10,6*X+10,1
50 NEXT
60 CLI
```

ON OFF

SYNTAXE: !ON
(!)OFF

Ces instructions rendent facultatif ou obligatoire le '!' devant les commandes du DOS.
OFF peut, parfois, accélérer certains programmes.

Sur l'Atmos, en mode OFF, les mots du DOS accessibles par les touches de fonctions seront automatiquement précédés du '!'.

A l'initialisation, l'Oric est en mode ON: '!' facultatif.

Remarque: Les commandes ON, FORMAT et PRINTER doivent dans tous les cas être précédées d'un '!', car elles commencent par un mot clé du BASIC.

FUNC

SYNTAXE: (!)FUNC ON
(!)FUNC OFF

Autorise ou inhibe les touches de fonctions (effet sur Atmos seulement).
Si vous devez enlever le connecteur de bus, un FUNC OFF est obligatoire.
A l'initialisation, l'ordinateur autorise les touches de fonctions.

Utilisation des touches de fonctions:
Atmos: 'Funct'+'Touche' ou
'Funct'+'Shift'+'Touche'

Deux touches de fonction jouent un rôle spécial:
-'Funct'+'Return' Génère un numéro de ligne (voir NUM).
-'Funct'+'DEL' Agit comme un 'DEL' simple, mais n'efface pas le caractère: Très pratique pour l'édition.

De plus, vous disposez d'un utilitaire DKEY qui vous permettra de vous familiariser avec les touches de fonction, de modifier leur disposition etc...

Pour y accéder, tapez DKEY ou dkey.

PRINTER

SYNTAXE: !PRINTER ON
!PRINTER OFF

Cette instruction dirige toutes les impressions vers l'écran, y compris les PRINT etc...

Attention: Dès que l'Oric affiche 'Ready', il passe en mode PRINTER OFF. Donc, pour utiliser cette instruction en mode direct, il faut mettre plusieurs instructions par ligne.

Exemples: !PRINTER ON:DIR
!PRINTER ON: LIST <=> LLIST

RESET

Cette instruction simule l'appui sur le poussoir 'RESET' des disquettes.
Elle lui est totalement équivalente.

Déroulement d'une initialisation:
A la mise sous tension, vous devez obligatoirement utiliser la disquette 'master' XL DOS.
Tant que vous n'éteignez pas votre Oric, vous pourrez 'booter' avec des disquettes 'esclaves' créées avec la commande INIT.

A chaque initialisation, l'XL DOS charge, s'il existe, le fichier 'BOOTUP.COM'.

SEI CLI

SYNTAXE: (!)SEI
(!)CLI

L'Oric utilise environ 20 pour cent de son temps pour gérer le clavier. Ces instructions permettent d'inhiber (SEI) ou de re-autoriser (CLI) cette gestion, permettant ainsi un gain de temps de 20 pour cent dans l'exécution des programmes.

Remarque: Si à la suite d'un SEI, le clavier se trouve bloqué, l'appui sur le poussoir 'RESET' de l'Oric

rendra la main.

FILE

SYNTAXE: (!)FILE NL

Précise le numéro de fichier qui sera pris si celui-ci n'est pas précisé.

Exemple: FILE 143

OPEN

Ouverture de fichier.

Direct: (!)OPEN "R",Nom de fichier,Longueur,NL

Chaîne: (!)OPEN "S",Longueur,NL

Disque: (!)OPEN "D",NL

Séquentiel: (!)OPEN "L",Nom de fichier,NL

Remarque: Les noms de fichiers ont des extensions spéciales, placées par l'XL DOS. Nom fichier a donc seulement 6 caractères utiles.

CLOSE

Fermeture des fichiers.

SYNTAXE: (!)CLOSE
Ferme tous les fichiers ouverts.

SYNTAXE: (!)CLOSE Type de fichier
Type: "S" ou "D" ou "L" ou "R".
Ferme tous les fichiers du type précisé.

SYNTAXE: (!)CLOSE NL
Ferme le fichier précisé.

NB: Ces deux dernières syntaxes peuvent être mixées à volonté.

Exemple: CLOSE 100,"D",12
Ferme tous les fichiers disque, les fichiers 100 et 12.

MSAVE / MLOAD

SYNTAXE: (!)MSAVE (#NL,)Nom du tableau
(!)MLOAD (#NL,)Nom du tableau

Sauvegarde et rappel de tableaux de chaînes ou de réels sur le disque.

Les fichiers ainsi créés sont exploitables de manière séquentielle.

Exemple: MSAVE AA$
MLOAD #4,I

Attention: Le tableau doit avoir été dimensionné. Si, lors d'un chargement, il se trouve sous dimensionné, il sera rempli, le reste du fichier n'étant pas chargé.

FSTART

SYNTAXE: (!)FSTART (#NL)

Place le pointeur du fichier séquentiel au début de celui-ci.

Exemple: FSTART 12

FJUMP

SYNTAXE: (!)FJUMP (#NL) nombre d'enregistrements

Saute le nombre d'enregistrements précisés sur le fichier spécifié.
Si le nombre est plus grand que le nombre d'enregistrements restants, le pointeur est positionné à la fin du fichier, sans générer d'erreur.

FEND

SYNTAXE: (!)FEND (#NL)

Place le pointeur de fichier séquentiel à la fin du fichier.

PUT

Direct: (!)PUT (#NL,)Numéro de la fiche
Ecrit le tampon dans l'enregistrement précisé.

Séquentiel: (!)PUT (#NL,)Liste de variables
Ecrit à la position courante du pointeur les variables.

Disque: (!)PUT (#NL,)Piste,Secteur,Lien piste,Lien secteur,Longueur utile
Ecrit le tampon sur le secteur précisé.

TAKE

Direct: (!)TAKE (#NL,)Numéro de fiche
Appel de la fiche dans le tampon du fichier considéré.

Séquentiel: (!)TAKE (#NL,)Liste de variables
Lit les variables considérées sur le fichier considéré.

Disque: (!)TAKE (#NL,)Piste,Secteur
Appelle dans le tampon le secteur précisé. Aucune vérification de l'existence du secteur n'est effectuée. Si le secteur n'existe pas, un FDERR=10 est généré.

FIELD

SYNTAXE: (!)FIELD (#NL,)Nom de champ TO Code(,Nom de champ TO Code)(,...)...

Définit les champs dans le tampon.
Le champ a 6 caractères significatifs et peut être un tableau à une dimension.

Code vaut: 0 :Pour un champ entier
255 :Pour un champ réel
1 à 254 :Pour un champ chaîne (soit la longueur de la chaîne)

Les champs sont séparés par des identificateurs de 1 octet, sauf pour les fichiers disques et chaînes. En tenir compte pour déterminer la longueur totale du champ.

LSET RSET

SYNTAXE: (!)LSET Nom de champ<Variable
(!)RSET Nom de champ<Variable

Permettent d'écrire dans les champs considérés. Totalement équivalentes pour les variables numériques. Les chaînes sont écrites en les justifiant à gauche (LSET) ou à droite (RSET).
Si la chaîne est plus courte que le champ, elle est complétée par des espaces.
Si la chaîne est plus longue que le champ, elle est tronquée à droite (LSET) ou à gauche (RSET).