

Les utilitaires

Paul Hogenboom

Une "boîte à outils" très complète!

Pour augmenter la puissance et les possibilités de son SED FLEX, Technical Systems Consultants (TSC) propose un certain nombre de programmes auxiliaires (les utilitaires). Parmi ceux-ci se trouvent quelques programmes auxiliaires dont la taille est inversement proportionnelle à leur utilité, à tel point que ce n'est pas sans raison que l'on peut affirmer qu'ils constituent une aide indispensable pour tout utilisateur de FLEX tant soit peu aventureux et sérieux. Etant donné l'espace disponible dans le cadre de cet article, nous ne pourrions nous pencher que sur quelques-uns des plus importants (les goûts et les couleurs se discutent). La documentation fournie avec la disquette comportant ces utilitaires est évidemment bien plus complète que nous ne pouvons l'être ici.

Pour la présentation de chaque programme nous adopterons un format commun qui sera le suivant dans l'ordre: description du programme et de son action, définition de la syntaxe, indication des valeurs ou adjonctions de défaut (la définition de ce terme sera donnée un peu plus loin), exemple, le résultat de l'exécution du programme. La syntaxe a pour fonction de définir très exactement les termes (ou caractères) à utiliser, et leur ordre pour demander l'exécution d'un programme (une instruction complexe en fait). Les termes de défaut peuvent être soit une valeur, soit un texte qu'il n'est pas nécessaire à l'utilisateur de préciser, le SED interprétant cette absence d'une manière bien définie. L'exemple type pour FLEX est l'extension .TXT (pour texte) que le logiciel ajoute "automatiquement" à tout fichier dont l'utilisateur n'a pas défini d'autre extension. Pour le reste de cet article, nous utiliserons dans nos exemples un fichier imaginaire que nous avons baptisé "PARAGR1.TXT".

Venons-en au fait:

FINDCMD

Cet utilitaire passe en revue un fichier texte pour y chercher une chaîne de caractères définie par l'utilisateur et lorsqu'il l'a trouvée, indique à l'écran ou sur imprimante, les lignes dans lesquelles il a trouvé la chaîne considérée. En fait, il s'agit là d'une instruction extrêmement utile lors de traitement de texte (ce type de logiciel étant dans la plupart des cas pourvu d'une telle fonction). Son existence hors programme de traitement de texte est très utile. On pourra utiliser cette instruction lorsqu'il faut rechercher dans un fichier texte (ou certain autres types de fichiers), un caractère, un mot, un chiffre ou même une phrase complète. Faire appel à FINDCMD prend sensiblement moins de temps que de charger un logiciel de traitement de texte de taille standard.

■ Syntaxe: FIND,Nom du fichier, chaîne de caractères à rechercher

■ Défaut(s): Le type du fichier est TXT

■ Exemple:
+++ FIND,PARAGR1,Disquette

■ Résultat: Il y a visualisation de toutes les lignes du fichier PARAGR1 dans lesquelles se trouve le mot "Disquette". On retrouve en outre le numéro d'ordre de chacune de ces lignes.

■ Produit:

+++ FIND,PARAGR1,Disquette
(Numéro de ligne)(1ère ligne dans laquelle se trouve "Disquette")
(Numéro de ligne)(2ème ligne dans laquelle se trouve "Disquette")
(etc. ...)

TOTAL STRING OCCURENCE IS XX

Comme l'illustre l'exemple, la liste des apparitions du mot recherché est suivie d'un message en indiquant le nombre d'occurrences.

WORDS

WORDSCMD est un autre utilitaire très pratique. Il ne nous est pas encore arrivé de rencontrer son équivalent dans un logiciel de traitement de texte sous FLEX. WORDS compte le nombre de mots et de lignes composant un texte, la connaissance de ces chiffres pouvant s'avérer extrêmement utile lorsqu'il faut respecter un nombre minimum (ou maximum) de mots (ou de lignes), ce dernier cas étant particulièrement intéressant pour la répartition du texte sur un certain nombre de pages.

■ Syntaxe: WORDS,Nom du fichier

■ Défaut(s): Le type du fichier est TXT

■ Exemple:

+++ WORDS,PARAGR1

■ Résultat: L'utilitaire affiche le nombre de mots et de lignes constituant le fichier (texte) cible: PARAGR1. Un espace séparant deux mots et le caractère "Retour à la ligne/ Curseur à gauche" sont les deux limiteurs de mots reconnus.

■ Produit:

+++ WORDS,PARAGR1
TOTAL WORD COUNT IS 1662
TOTAL LINE COUNT IS 144

TYPOS

TYPOSCMD est une version de luxe de l'instruction précédente. Elle indique d'une part le nombre de mots que comporte le fichier cible et génère en outre une liste indiquant la fréquence d'apparition de tous les mots constituant le fichier concerné. TYPOS permet ainsi de vérifier la bonne orthographe de tous les mots utilisés, de s'assurer de l'absence de termes techniquement obsolètes ou d'éviter les répétitions trop fréquentes. Comme certains mots (articles, pronoms et autres conjonctions) apparaissent bien plus fréquemment que d'autres, il faut indiquer le nombre maximal de répétition admis. En l'absence d'indication de "fréquence" maximale, le logiciel retient le chiffre "3" comme limite.

■ Syntaxe: TYPOS,Nom du fichier, fréquence maximale

■ Défaut(s): Le type de fichier est TXT; la fréquence de répétition maximale est 3 (en l'absence d'indication de limite, tous les mots présents dans le texte trois fois ou moins

sont indiqués).

■ Exemple: TYPOS,PARAGR1,5

■ Résultat: Une liste de tous les mots présents cinq, quatre, trois, deux ou une fois dans le fichier texte "PARAGR1". L'affichage indique la fréquence devant le mot considéré. Cette liste est classée par fréquence (et par ordre alphabétique dans chaque catégorie de fréquence).

■ Produit: Comme une telle liste est extrêmement longue, nous n'indiquerons pas ici le produit de l'instruction TYPOS. Vous pouvez sans doute vous imaginer ce qu'elle pourrait être.

SPLIT

C'est lorsque l'on travaille avec des textes de longueur importante que SPLITCMD prouve son utilité. Comme l'écran ne peut visualiser plus de 25 lignes, il est aisé de perdre le fil avec ce genre de textes. La division d'un tel texte en plusieurs petits morceaux peut faciliter la tâche de l'utilisateur. L'utilitaire SPLITCMD divise un fichier texte en deux parties, le point de séparation étant défini par l'utilisateur sous la forme d'un numéro de ligne.

■ Syntaxe: SPLIT,Nom du fichier d'origine,Nom du premier nouveau fichier,Nom du second nouveau fichier,Numéro de la ligne où doit avoir lieu la césure.

■ Défaut(s): Le type de fichier est TXT.

■ Exemple:
+++ SPLIT,PARAGR1,PARTIE1,
PARTIE2,100

■ Résultat: Le fichier d'origine PARAGR1 est découpé en deux nouveaux fichiers dont le premier sera baptisé PARTIE1 et comportera les lignes 1 à 99 incluse, le second PARTIE2 comprendra les lignes à compter de 100 jusqu'à la fin. Le fichier d'origine PARAGR1.TXT ne subit pas de modification au cours de cette opération.

LOW-UP, UP-LOW

LOW-UPCMD et UP-LOWCMD sont deux routines qui effectuent la conversion des minuscules en majuscules et inversement. Cette conversion peut être nécessaire par exemple, quand un programme écrit dans un BASIC admettant les deux types de lettres doit être interprété par un autre BASIC qui n'admet que les majuscules.

■ Syntaxe: LOW-UP,Nom du fichier d'origine avec majuscules, Nom du nouveau fichier comportant des majuscules exclusivement.

■ Défaut(s): Le type de fichier est TXT.

■ Exemple:
+++ LOW-UP,MINUSC,MAJUSC

■ Résultat: A partir du fichier MINUSC qui comportait des majuscules et des minuscules est généré un fichier MAJUSC composé de majuscules uniquement.

L'exécution de la routine UP-LOW produit bien évidemment le résultat inverse.

DUMP

DUMPCMD visualise, à l'écran ou sur imprimante, le contenu d'un (ou de plusieurs) secteur(s) d'une disquette. Les premiers éléments affichés sont les adresses de la piste et du secteur, le contenu du secteur concerné étant ensuite visualisé à raison de 16 octets par ligne, tant en format hexadécimal qu'en format ASCII. Cette routine est extrêmement précieuse en cas de détérioration d'une disquette que l'on pourra ainsi réparer à condition de posséder un listing sur papier du secteur endommagé.

■ Syntaxe: DUMP,Nom du fichier, Type du fichier

■ Défaut(s): Le type du fichier est BIN (binaire).

■ Exemple:
+++ DUMP,SCRHELP.SYS

■ Résultat: On voit apparaître à l'écran l'adresse de début du fichier SCRHELP.SYS (numéros de la piste et du secteur en format hexadécimal). On aperçoit ensuite le contenu du secteur en 16 lignes de 16 octets (traduits en caractères ASCII sur la seconde partie de l'écran). Pour info, chaque secteur comprend 256 octets. Tous les secteurs faisant partie du fichier à visualiser apparaissent ainsi à l'écran.

■ Produit: Voir figure 1

CHECK

CHECKCMD compare les contenus de deux fichiers. Lorsque cette opération est terminée, le résultat de la comparaison est donné sous la forme de l'un des deux messages



CHECKED DO NOT
THE FILES CHECKED ARE IDENTICAL.

Cette routine est très pratique lorsque l'on se trouve en présence de deux fichiers de noms différents dont on suppose que l'un est la copie de l'autre.

■ Syntaxe: CHECK,Nom du fichier1, Nom du fichier2

■ Défaut(s): Le type du fichier est TXT.

■ Exemple:

+++CHECK,PARAGRI.TXT,PARAGRI.BAK

■ Résultat: Le contenu du fichier PARAGRI est comparé à son fichier de sauvegarde (BAK = Back-up).

■ Produit:

+++CHECK,PARAGRI.TXT,PARAGRI.BAK
THE FILES CHECKED ARE IDENTICAL

MAP

MAPCMD est un programme permettant de retrouver les adresses de chargement et les domaines occupés par les fichiers de commande. Il indique à quelle adresse est implanté un fichier commande et à quelle adresse prend place son exécution, (cette dernière adresse est également appelée adresse de transfert). L'utilisation de MAP se justifie tout-particulièrement de concert avec l'instruction SAVE qui permet d'écrire sur disquette, sous n'importe quel nom, le contenu d'un bloc de mémoire choisi.

■ Syntaxe: MAP,Nom du fichier

■ Défaut(s): Le type du fichier est BIN.

■ Exemple:

+++MAP,MAPCMD

■ Résultat: On voit apparaître à l'écran les adresses de début et de fin du domaine mémoire dans lequel est chargé MAPCMD, suivies de l'adresse de transfert

■ Produit:

+++MAP,MAPCMD
C100-C126
C100-C1E8
C100

FREE

FREECMD indique la place disponible sur la disquette, place donnée en nombre de secteurs et en Koctets.

■ Syntaxe: FREE X

(X = numéro d'ordre du lecteur concerné)

■ Défaut(s): Le lecteur est le lecteur 0.

■ Exemple:

+++FREE

■ Résultat: On voit apparaître à l'écran le message suivant:
SECTORS REMAINING = X
APPROXIMATE KILOBYTES = YY

■ Produit:

+++FREE
SECTORS REMAINING = 243
APPROXIMATE KILOBYTES = 61

TEST

TESTCMD permet d'épargner de gros déboires à l'utilisateur de disquettes de moindre qualité. Tous les secteurs de la disquette sont passés en revue pour vérifier leur intégrité; si la disquette comporte de mauvais secteurs, le message BAD SECTOR est affiché à l'écran avec le numéro hexadécimal du (ou des) secteur(s) concerné(s).

■ Syntaxe: TEST,Numéro du lecteur

■ Défaut(s): Le numéro est celui du lecteur défini comme lecteur de travail par l'instruction ASN (voir l'article sur le SED FLEX).

■ Exemple: TEST,0

■ Résultat: La disquette présente dans le lecteur 0 subit un test rigoureux; la détection d'une erreur est signalée par le message BAD SECTOR AT.... Si l'on demande de

```

03 20
03 21 00 01 43 6F 6D 6D 61 6E 64 73 20 69 6E 20 _!__Commands in
74 68 65 20 63 6F 6D 6D 61 6E 64 20 6D 6F 64 65 the command mode
3A 0D 0D 50 41 47 45 3D 4E 09 0B 53 65 74 20 6E :__PAGE=N__Set n
65 77 20 70 61 67 65 6C 65 6E 67 74 68 2E OD 4C ew pagelength. L
49 4E 45 3D 4E 09 0B 53 65 74 20 6E 65 77 20 6C INE=N__Set new l
69 6E 65 6C 65 6E 67 74 68 2E OD 53 54 41 54 55 inelength. STATU
53 3D 54 52 55 45 09 06 45 6E 61 62 6C 65 20 73 S=TRUE__Enable s
74 61 74 75 73 20 6C 69 6E 65 20 69 6E 20 74 65 tatus line in te
78 74 20 6D 6F 64 65 2E OD 53 54 41 54 55 53 3D xt mode. STATUS=
46 41 4C 53 45 09 05 44 69 73 61 62 6C 65 20 73 FALSE__Disable s
74 61 74 75 73 20 6C 69 6E 65 2E OD 49 4E 53 45 tatus line. INSE
52 54 3D 54 52 55 45 09 06 45 6E 61 62 6C 65 20 RT=TRUE__Enable
61 75 74 6F 20 66 69 65 6C 64 69 6E 67 2E OD 49 auto fielding. I
4E 53 45 52 54 3D 46 41 4C 53 45 09 05 44 69 73 NSERT=FALSE__Dis
61 62 6C 65 20 61 75 74 6F 20 66 69 65 6C 64 69 6E 67 2E OD 49 able auto fieldi
6E 67 2E OD 46 49 45 4C 44 3D 4E 09 0A 44 65 66 ng. FIELD=N__Def

03 21
03 22 00 02 69 6E 65 20 66 69 65 6C 64 20 63 68 _"__line field ch
61 72 61 63 74 65 72 20 61 73 20 64 65 63 69 6D aracter as decim
61 6C 20 6E 75 6D 62 65 72 2E OD 46 49 45 4C 44 al number. FIELD
3D 27 43 09 09 44 65 66 69 6E 65 20 66 69 65 6C ='C__Define fiel
64 20 63 68 61 72 61 63 74 65 72 20 61 73 20 41 d character as A
53 43 49 49 20 63 68 61 72 2E OD 43 4F 4D 4D 45 SCII char. COMME
4E 54 3D 2A 09 08 44 65 66 69 6E 65 20 63 6F 6D NT=*__Define com
6D 65 6E 74 20 63 68 61 72 61 63 74 65 72 2E OD ment character.
57 49 4C 44 3D 3F 09 0B 44 65 66 69 6E 65 20 77 WILD=?__Define w
69 6C 64 20 73 65 61 72 63 68 20 63 68 61 72 61 d search chara
63 74 65 72 2E OD 46 4F 4C 44 3D 54 52 55 45 09 cter. FOLD=TRUE
08 53 65 74 20 75 70 70 65 72 63 61 73 65 20 6F _Set uppercase o
6E 6C 79 2E OD 46 4F 4C 44 3D 46 41 4C 53 45 09 nly. FOLD=FALSE
07 55 70 70 65 72 2F 6C 6F 77 65 72 63 61 73 65 _Upper/lowercase
2E OD 53 45 41 52 43 48 46 4F 4C 44 3D 54 52 55 _SEARCHFOLD=TRU
45 09 02 53 65 61 72 63 68 20 75 70 70 65 72 20 E__Search upper

03 22
03 23 00 03 6F 72 20 6C 6F 77 65 72 63 61 73 65 _£__or lowercase
2E OD 53 45 41 52 43 48 46 4F 4C 44 3D 46 41 4C _SEARCHFOLD=FAL
53 45 20 53 65 61 72 63 68 20 66 6F 72 20 65 78 SE Search for ex
61 63 74 20 6D 61 74 63 68 2E OD 41 55 54 4F 3D act match. AUTO=
54 52 55 45 09 08 41 75 74 6F 20 6E 75 6D 62 65 TRUE__Auto numbe
72 69 6E 67 20 6F 6E 2E OD 41 55 54 4F 3D 46 41 ring on. AUTO=FA
4C 53 45 09 07 41 75 74 6F 20 6E 75 6D 62 65 72 LSE__Auto number
69 6E 67 20 6F 66 66 2E 5C OD OD 4F 6E 65 20 6C ing off. \__One l
65 74 74 65 72 20 63 6F 6D 6D 61 6E 64 73 3A OD etter commands:
OD 41 20 6F 72 20 51 09 0B 41 62 6F 72 74 20 6F _A or Q__Abort o
72 20 51 75 69 74 2E 20 4E 6F 20 73 61 76 65 2E r Quit. No save.
OD 53 09 10 53 61 76 65 20 61 6C 6C 20 74 65 78 _S__Save all tex
74 20 61 6E 64 20 72 65 74 75 72 6E 20 74 6F 20 t and return to
46 4C 45 58 2E OD 47 09 10 47 6F 20 65 78 65 63 FLEX. G__Go exec
75 74 65 20 70 72 65 64 65 66 69 6E 65 64 20 47 ute predefined G
4F 20 63 6F 6D 6D 61 6E 64 2E OD 4E 5B 6D 5D 09 O command. N[m]_

```


tester une disquette dans un lecteur n'existant pas, on aura affichage du message suivant: **ILLEGAL DRIVE NUMBER.**

■ **Produit:**

+++TEST,0
BAD SECTOR AT NN-MM
BAD SECTOR AT XX-YY
etc...

TEST COMPLETED

soit:

+++TEST,0
TEST COMPLETED

C'est bien évidemment ce second message que l'on souhaite voir s'afficher à l'écran: une disquette parfaite.

RPT

RPTCMD simplifie la tâche lorsqu'il faut qu'un programme soit exécuté à plusieurs reprises. Normalement, il faudrait à chaque fois entrer à nouveau la commande à exécuter. Cela est bien évidemment possible, mais l'efficacité en prend un coup (à quoi cela sert-il d'avoir un ordinateur?). RPT (de repeat, vous l'avez deviné) permet de faire exécuter une commande le nombre de fois que l'on veut.

■ **Syntaxe:** RPT,Nombre d'exécutions désiré,commande à exécuter

■ **Défaut(s):** sans objet

■ **Exemple:** RPT,3,P LIST PARAGR1.TXT

■ **Résultat:** Le contenu du fichier PARAGR1 est imprimé en triple exemplaire sur l'imprimante.

PDEL

PDELCMD est l'abréviation de "prompting delete", effacement par acquiescement en quelque sorte. Dans le cas de la commande d'effacement DELETE standard, il faut indiquer le nom du fichier à effacer. Ici les choses se passent différemment. Sur l'écran apparaissent les noms de tous les fichiers présents dans le répertoire de la disquette dans l'ordre correspondant à ce celui de ce dernier. Le nom de chaque fichier est suivi par un point d'interrogation auquel on peut répondre soit par Y (yes), N (no) ou

Retour à la ligne.

Un retour à la ligne fait sortir du programme PDEL; le système revient en mode commande (+++). Important: avant de procéder à l'effacement, PDEL ne pose pas la question de confiance "ARE YOU SURE" (contrairement à ce que fait DELETE.CMD), ce qui signifie qu'une action sur Y provoque une action immédiate! Il est possible de limiter à un certain type de fichier les fichiers apparaissant à l'écran (.TXT par exemple). Il est possible de cette manière de filtrer, par combinaisons de lettres interposées les fichiers à effacer.

■ **Syntaxe:** PDEL,Numéro de lecteur,(critères de sélection)

■ **Défaut(s):** sans objet

■ **Exemple:** PDEL,0,.TXT

■ **Résultat:** Les noms de tous les fichiers texte de la disquette présente dans le lecteur 0 apparaissent l'un à la suite de l'autre à l'écran. Après chaque nom, il faut entrer soit un Y, soit un N, soit un Retour à la ligne. Après une action sur Y ou N, le curseur passe au fichier suivant, après un Retour à la ligne ou le dernier fichier, le système revient en mode commande (+++).

MEMDUMP

MEMDUMPCMD génère l'apparition à l'écran en blocs de 256 octets du contenu de la mémoire sous format hexadécimal. A la suite du premier bloc on voit apparaître trois lettres donnant le choix entre bloc suivant (F = forward), bloc précédent (B = backward) et sortie du programme (Retour à la ligne). Dans le premier cas, on voit apparaître le bloc de 256 octets suivant, dans le second le bloc précédent et dans le dernier cas on retourne en mode commande (+++).

■ **Syntaxe:** MEMDUMP,Adresse de début en hexadécimal

■ **Défaut(s):** En l'absence d'adresse de début, le programme suppose qu'il s'agit de 0000.

■ **Exemple:** MEMDUMP,F000

■ **Résultat:** Sur l'écran apparaît en format hexadécimal le contenu de la mémoire du domaine compris entre F000 et F0FF.

MEMTEST

MEMTESTCMD est un programme

permettant, comme l'indiquent leurs son nom, de tester le fonctionnement de la mémoire, une fonction extrêmement utile lorsque l'on a des doutes à ce sujet.

■ **Syntaxe:** MEMTEST,Adresse de début,Adresse de fin.

■ **Défaut(s):** sans objet

■ **Exemple:** MEMTEST,0,01FF

■ **Résultat:** Le système écrit un nombre aléatoire dans chacune des cellules de mémoire du domaine compris entre 0000 et 01FF; il procède ensuite à la lecture de ces nombres et les compare au nombre d'origine, cette procédure se répétant un certain nombre de fois. Le test d'un bloc de 4 Koctets dure quelque 60 minutes. Il n'est bien évidemment pas possible de tester le domaine mémoire dans lequel réside le programme MEMTEST sous peine de "planter" le système. Pour éviter ce problème, on pourra, à l'aide du programme MAP, localiser sans peine l'endroit où réside MEMTEST. A la fin du processus de test, si tout s'est bien passé, l'écran affiche un "I" victorieux. La seule solution pour interrompre l'exécution de MEMTEST est d'actionner la touche RESET.

Comme nous le signalions en début d'article, nous ne disposons pas de suffisamment d'espace pour passer en revue tous les programmes utilitaires de FLEX, surtout qu'il en naît des nouveaux chaque jour. Pour terminer cet article nous indiquons dans l'ordre alphabétique les différents utilitaires décrits dans cet article:

CHECK
DUMP
FIND
FREE
LOW-UP
MAP
MEMDUMP
MEMTEST
PDEL
RPT
SPLIT
TEST
UP-LOW
WORDS

Plusieurs de nos annonceurs proposent de nombreux logiciels tournant sous Flex. Voir entre autres aux pages 102 et 103.