

Préparation des programmes

Bull DPS 7

Guide de l'utilisateur

FSE - Editeur plein écran

GCOS 7000

Logiciel

Sujet : Ce guide est une présentation générale de l'éditeur plein écran (FSE) utilisé sous IOF (Interactive Operation Facility).

Observations : La révision 1 est valable pour tous les utilisateurs de GCOS 7 V3. Elle remplace la révision 0 qui reste valable pour les utilisateurs des versions V1 et V2 de GCOS 7.

Version du logiciel : GCOS 7-XLS Versions V3A, V3B
GCOS 7-LS Version V3A
GCOS 7-MS Version V3A
GCOS 7-ES Version V3A
GCOS 7-V5
GCOS 7-V6

Date : Décembre 1991

Bull Electronics Angers S.A.
CEDOC
Atelier de reprographie
331, Avenue Patton
49004 ANGERS Cedex 01
FRANCE

Bull HN Information Systems Inc.
Publication Order Entry
FAX: (508) 294-7411
MA02/423S
Technology Park
Billerica, MA 01821
U.S.A.

Copyright © Bull S.A., 1991

Toutes les marques citées sont la propriété de leurs titulaires respectifs.

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve à la fin du présent manuel.

La loi du 11 mars 1957, complétée par la loi du 3 juillet 1985, interdit les copies ou reproductions destinées à une utilisation collective. Toute représentation ou reproduction intégrale ou partielle faite par quelque procédé que ce soit, sans consentement de l'auteur ou de ses ayants cause, est illicite et constitue une contrefaçon sanctionnée par les articles 425 et suivants du code pénal.

Ce document est fourni à titre d'information seulement. Il n'engage pas la responsabilité de Bull S.A. en cas de dommages résultant de son application. Des corrections ou modifications au contenu de ce document peuvent intervenir sans préavis ; des mises à jour ultérieures les signaleront éventuellement aux destinataires.

Préface

OBJET DU MANUEL

Ce guide présente toutes les informations nécessaires pour utiliser l'éditeur plein écran (FSE - Full Screen Editor) sous IOF (Interactive Operation Facility).

UTILISATEURS CONCERNES

Ce guide s'adresse à tous les utilisateurs de GCOS 7 qui ont besoin des fonctions de FSE.

STRUCTURE

Ce guide utilisateur est divisé en cinq chapitres :

Le chapitre 1 est une introduction à FSE et présente les concepts sur lesquels il repose. On y trouvera également une brève description des commandes FSE disponibles.

Le chapitre 2 fournit une description détaillée de chaque commande, avec des exemples. Par souci de commodité, les commandes sont présentées en ordre alphabétique.

Le chapitre 3 présente les principales caractéristiques d'une session MODIFY.

Le chapitre 4 contient une description des demandes utilisables dans le cadre d'une session MODIFY. Les demandes sont présentées en ordre alphabétique.

Le chapitre 5 traite des fonctions offertes en mode écran.

Pour avoir une bonne compréhension de FSE, il est recommandé de lire la totalité du chapitre 1 ainsi que les parties des chapitres 2 et 3 correspondant aux commandes et demandes dont vous avez besoin.

BIBLIOGRAPHIE

Les références suivantes peuvent être consultées mais ne sont pas indispensables à la compréhension de l'information fournie dans ce guide.

<i>IOF : Manuel de référence de l'utilisateur</i>	
<i>Vol. 1 - Introduction à IOF</i>	47 F2 01UJ
<i>Vol. 2 - Commandes GCL</i>	47 F2 02UJ
<i>Vol. 3 - Commandes processeurs</i>	47 F2 03UJ
<i>Vol. 4 - Compléments</i>	47 F2 04UJ
<i>IOF Terminal User's Reference manual :</i>	
<i>Introduction to IOF</i>	47 A2 01UJ
<i>GCL Commands</i>	47 A2 02UJ
<i>Processor Commands</i>	47 A2 03UJ
<i>Appendices</i>	47 A2 04UJ
<i>Aide à la mise au point</i>	47 F2 15UP
<i>Program Checkout facility</i>	47 A2 15UP
<i>Maintenance des bibliothèques - Manuel de référence</i>	47 F2 01UP
<i>Maintain Library Reference manual</i>	47 A2 01UP
<i>Maintenance des bibliothèques - Guide de l'utilisateur</i>	47 F2 02UP
<i>Maintain Library User Guide</i>	47 A2 02UP

"Maintain Library" est aussi connu sous le nom de LIBMAINT.

Dans ce manuel, les références sont citées sous forme abrégée (4 derniers caractères alphanumériques). Pour plus de précisions (disponibilité du manuel, numéro de révision, indice de mise à jour), veuillez consulter le "Catalogue de la documentation" et le fascicule "Documents nouveaux" qui portent tous deux la référence 00 F4 7210.

CONVENTIONS D'ECRITURE

Les conventions d'écriture utilisées dans ce manuel pour la syntaxe des commandes/demandes sont les suivantes:

ELEMENT:	Les majuscules indiquent un mot-clé, à introduire tel quel.
élément :	Les minuscules indiquent une valeur à fournir par l'utilisateur.
[élément] :	Les crochets indiquent un élément facultatif.
{élément 1}	Une colonne d'éléments entre accolades indique que l'un d'eux doit être sélectionné. La valeur implicite (éventuelle) est soulignée (ici, élément 3).
{élément 2}	
{ <u>élément 3</u> }	
... :	Les points de suspension indiquent que l'élément qui précède peut être répété une ou plusieurs fois.

Table des matières

1.	Généralités et concepts de base	1-1
1.1	GENERALITES	1-1
1.2	BIBLIOTHEQUES DE TRAVAIL	1-2
1.3	ENVIRONNEMENT	1-3
1.4	APPEL DE FSE	1-4
1.4.1	Appel de FSE au niveau système	1-4
1.4.1.1	Utilisation des menus.....	1-4
1.4.1.2	Utilisation du mode ligne.....	1-5
1.4.1.3	Paramètre LIB.....	1-5
1.4.2	Appel de FSE à partir d'un autre processeur	1-6
1.4.2.1	Utilisation des menus.....	1-6
1.4.2.2	Utilisation du mode ligne.....	1-6
1.5	UTILISATION DE FSE	1-7
1.5.1	Noms d'unité de bibliothèque	1-7
1.5.2	Commandes FSE	1-7
1.6	GRILLES PROPRES AUX DIFFERENTS LANGAGES	1-10

2.	Description des commandes	2-1
2.1	COMMANDE COPY	2-2
2.2	COMMANDE CREATE	2-5
2.3	COMMANDE DELETE	2-8
2.4	COMMANDE DISPLAY	2-11
2.5	COMMANDE INLIBN.....	2-12
2.6	COMMANDE LIB.....	2-15
2.7	COMMANDE LIST	2-17
2.8	COMMANDE MODIFY.....	2-20
2.9	COMMANDE PRINT.....	2-22
2.10	COMMANDE QUIT.....	2-25
2.11	COMMANDE RENAME	2-26
2.12	COMMANDE RENUMBER	2-27
3.	Session MODIFY	3-1
3.1	PRESENTATION.....	3-1
3.2	DESCRIPTION DES GRILLES.....	3-2
3.2.1	Grille principale	3-2
3.2.1.1	Zone en-tête	3-3
3.2.1.2	Zone texte	3-4
3.2.1.3	Zone "request"	3-4
3.2.1.4	Opérations	3-4

Table des Matières

3.2.2	Grille de guidage	3-6
3.2.2.1	Opérations	3-7
3.3	VISUALISATION DE TEXTE	3-9
3.4	SAUT DE PAGE	3-10
3.5	INTRODUCTION DES DEMANDES	3-11
3.5.1	Demandes "ligne"	3-11
3.5.2	Mode libre	3-12
3.6	NUMEROTATION DES LIGNES	3-14
3.7	ADRESSAGE DE LIGNE	3-16
3.7.1	Adressage par numéro de ligne	3-16
3.7.2	Adressage par application d'un incrément	3-17
3.7.3	Adressage par identification d'un profil	3-17
3.7.3.1	Expression régulière	3-18
3.7.3.2	Chaîne régulière	3-18
3.7.4	Combinaison des méthodes d'adressage	3-20
3.7.4.1	Adressage d'une plage de lignes	3-20
3.7.4.2	Adressage d'un ensemble de lignes	3-21
3.7.5	Règles de spécification des adresses	3-23
3.8	MODE SAISIE	3-24
3.9	MODE FORMATE	3-28
3.10	CHANGEMENT DES CARACTERES DE COMMANDE DE FSE ('^', '\$' ET '[').	3-29
3.11	TAMPON	3-30
3.11.1	Nom de tampon	3-30
3.11.2	Association d'un nom d'unité à un tampon.	3-31
3.12	MECANISME DE PROTECTION	3-32

3.13	FENETRE DE VISUALISATION	3-33
3.13.1	Zone en-tête	3-34
3.13.2	Zone texte	3-34
3.13.3	Zone demande (request)	3-35
3.14	CONTEXTES	3-36
3.14.1	Identificateur de contexte	3-36
3.14.2	Mise à jour de contexte.	3-37
3.15	MACROS (EXECUTION DU CONTENU D'UN TAMPON)	3-39
3.16	SEQUENCE DE LANCEMENT	3-42
3.17	TRAITEMENT DES CARACTERES NON GRAPHIQUES	3-45
3.18	FONCTIONS D'AFFICHAGE, D'OCCULTATION ET DE MISE EN EVIDENCE	3-46
3.19	DEMANDES DE MODIFY	3-47
4.	DEMANDES "LIGNE"	4-1
4.1	DEMANDE A	4-2
4.2	DEMANDE AB	4-3
4.3	DEMANDE ADD	4-4
4.4	DEMANDE AF	4-5
4.5	DEMANDE ASG	4-6
4.6	DEMANDE BNDS	4-7
4.7	DEMANDE C	4-8

Table des Matières

4.8	DEMANDE CC	4-9
4.9	DEMANDE CF	4-10
4.10	DEMANDE CLB	4-11
4.11	DEMANDE CLC	4-12
4.12	DEMANDE COM	4-13
4.13	DEMANDE CW	4-14
4.14	DEMANDE D	4-15
4.15	DEMANDE DISP	4-17
4.16	DEMANDE DQ	4-20
4.17	DEMANDE EA	4-21
4.18	DEMANDE EI	4-22
4.19	DEMANDE EXP	4-23
4.20	DEMANDE FC	4-24
4.21	DEMANDE H	4-25
4.22	DEMANDE HL	4-28
4.23	DEMANDE I	4-29
4.24	DEMANDE IB	4-30

4.25	DEMANDE IF	4-31
4.26	DEMANDES IL1, IL2, IL3	4-32
4.27	DEMANDE IND.....	4-33
4.28	DEMANDE K.....	4-34
4.29	DEMANDE KA.....	4-35
4.30	DEMANDE KI	4-36
4.31	DEMANDE KR.....	4-37
4.32	DEMANDE L	4-38
4.33	DEMANDE LE	4-40
4.34	DEMANDE LIB	4-41
4.35	DEMANDE LO.....	4-42
4.36	DEMANDE M.....	4-44
4.37	DEMANDE MA	4-45
4.38	DEMANDE MI.....	4-46
4.39	DEMANDE MR	4-47
4.40	DEMANDE NAME	4-48
4.41	DEMANDE NOV	4-50
4.42	DEMANDE OW.....	4-51

Table des Matières

4.43	DEMANDE PF	4-52
4.44	DEMANDE QC	4-53
4.45	DEMANDES QUIT, Q , /	4-54
4.46	DEMANDE R	4-55
4.47	DEMANDE RA	4-56
4.48	DEMANDE RB	4-58
4.49	DEMANDE REM	4-59
4.50	DEMANDE RI	4-60
4.51	DEMANDE S	4-62
4.52	DEMANDE SB	4-66
4.53	DEMANDE SE	4-67
4.54	DEMANDE SEND	4-68
4.55	DEMANDE SP	4-69
4.56	DEMANDE SQ	4-70
4.57	DEMANDE STP	4-71
4.58	DEMANDE SV	4-72
4.59	DEMANDE SVQ	4-74
4.60	DEMANDE TOP	4-75
4.61	DEMANDE UC	4-76

4.62	DEMANDE UH.....	4-77
4.63	DEMANDE UL.....	4-78
4.64	DEMANDE ULF.....	4-81
4.65	DEMANDE UP.....	4-82
4.66	DEMANDE USTP.....	4-84
4.67	DEMANDE ?.....	4-85
4.68	DEMANDE =.....	4-86
4.69	DEMANDES > ET <.....	4-87
4.70	DEMANDE <NOM DE TAMPON>.....	4-88
5.	MODE ECRAN.....	5-1
5.1	PRESENTATION.....	5-1
5.2	DESCRIPTION DES GRILLES.....	5-2
5.3	OPTIONS DU MODE ECRAN.....	5-3
5.3.1	Mise à jour du texte.....	5-4
5.3.2	Demands "écran".....	5-4
5.3.2.1	SYNTAXE.....	5-5
5.3.2.2	DEMANDES "ECRAN" PORTANT SUR UNE SEULE LIGNE.....	5-6
5.3.2.3	DEMANDES PORTANT SUR UN BLOC DE LIGNES.....	5-7
5.3.3	Demands "ligne".....	5-9

Illustrations

Figures

1-1	Bibliothèques de travail.....	1-2
1-2	Environnement FSE.....	1-3

1. Généralités et concepts de base

Ce chapitre est une introduction à l'éditeur plein écran (FSE - Full Screen Editor) et expose les concepts sur lesquels il repose, tels que les bibliothèques de travail, le niveau d'appel (à partir du système ou à partir d'un processeur intermédiaire), la méthode d'appel (en mode menu ou en mode ligne). On y trouvera également une brève description des commandes FSE disponibles.

1.1 GENERALITES

FSE est un processeur dont la principale fonction est de gérer (c'est-à-dire créer, modifier etc.) des unités de bibliothèque origine (SL). Il peut aussi changer le nom et le numéro d'unités de bibliothèque origine, ou en copier le contenu d'une bibliothèque dans une autre. Les modifications sont effectuées sur une image de l'unité concernée, rangée temporairement dans une zone appelée "tampon". Quand le contenu de cet espace de travail (tampon) est jugé satisfaisant, il est possible de le transférer dans une unité de la bibliothèque SL de sortie (voir ci-dessous "Bibliothèques de travail").

1.2 BIBLIOTHEQUES DE TRAVAIL

FSE peut travailler simultanément avec quatre bibliothèques SL au maximum, dont une seule en sortie.

Les opérations de lecture peuvent s'effectuer dans les bibliothèques d'entrée et dans la bibliothèque de sortie, alors que les opérations d'écriture ne peuvent s'effectuer que dans la bibliothèque de sortie.

Les bibliothèques d'entrée peuvent être affectées, avant de lancer FSE, au moyen de la commande système MWINLIB. La bibliothèque de sortie peut être affectée, avant de lancer FSE, au moyen de la commande système MWLIB ou du paramètre LIB de la commande système FSE (voir ci-dessous "Appel de FSE"). Lorsque FSE est appelé au niveau système (c'est-à-dire en tant que processeur indépendant), les bibliothèques peuvent également être affectées au moyen des commandes LIB et INLIBn de FSE.

Pour FSE, les bibliothèques sont désignées par leur nom symbolique : LIB, INLIB1, INLIB2 et INLIB3.

Toute référence à une bibliothèque non affectée provoque la visualisation d'un message d'erreur.

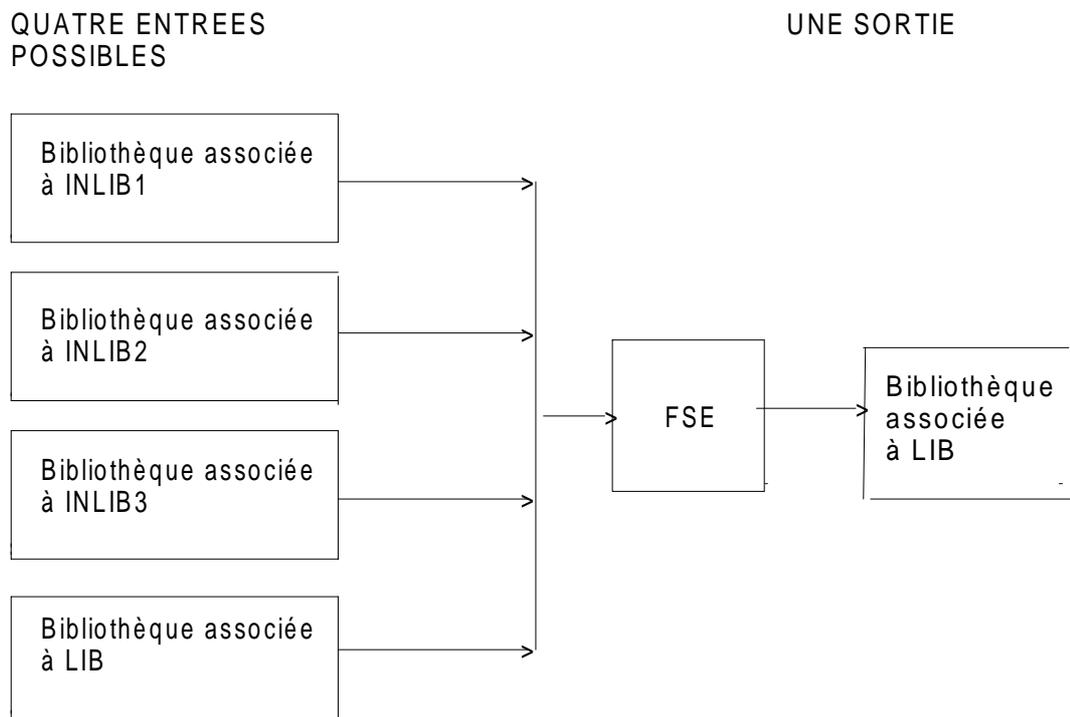


Figure 1-1. Bibliothèques de travail

1.3 ENVIRONNEMENT

Il est possible d'exécuter FSE sous IOF, à partir de terminaux disposant de FORMS, soit en tant que processeur indépendant soit en tant que processeur subordonné à d'autres processeurs, qui sont :

- MAINTAIN_COMMAND (Gestion de commandes)
- MAINTAIN_FORM (Création et gestion de grilles)
- MAINTAIN_LIBRARY (Maintenance de bibliothèques SL)
- PCF (Aide à la mise au point)

Les opérations à exécuter dans le cadre de FSE sont définies au moyen d'une séquence de commandes FSE. Une documentation en ligne, sous forme de textes HELP, est à la disposition de l'utilisateur à tous les niveaux d'exécution. Ces textes contiennent des extraits des informations de ce guide.

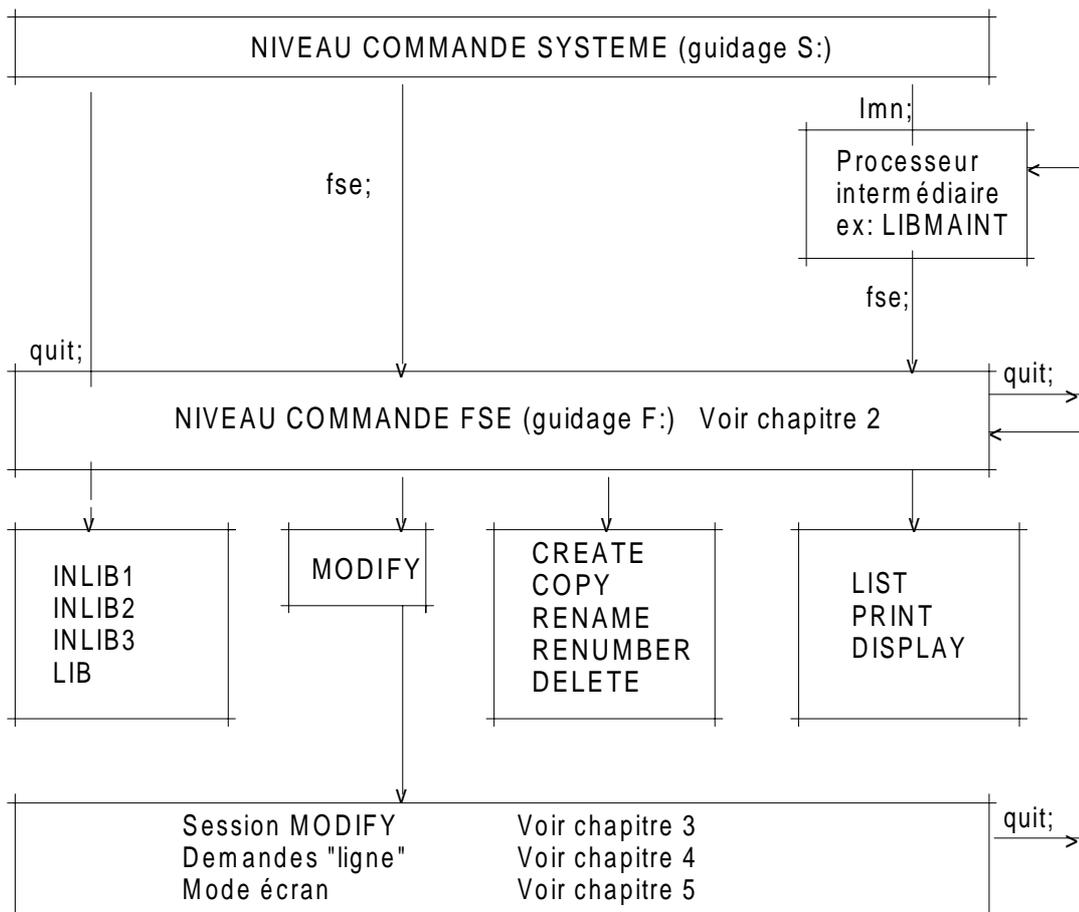


Figure 1-2. Environnement FSE

1.4 APPEL DE FSE

Comme il a déjà été indiqué, FSE peut être appelé soit en tant que processeur indépendant, au niveau système, soit à partir d'un autre processeur, au niveau commande.

Dans les deux cas, il y a deux méthodes possibles pour appeler FSE et exécuter des commandes. La première consiste à utiliser les menus en sélectionnant sur une liste d'opérations possibles l'option **Error! Reference source not found.** La deuxième consiste à introduire en mode ligne la commande d'appel de FSE. Les deux méthodes sont décrites ci-dessous.

1.4.1 Appel de FSE au niveau système

1.4.1.1 Utilisation des menus

Des précisions sur l'utilisation des menus et l'affichage des textes HELP peuvent être obtenues en introduisant un point d'interrogation (?) dans la zone OPERATIONS du menu TUTORIAL. Pour afficher le menu TUTORIAL, introduire TUTORIAL? au niveau système. Ces renseignements peuvent également être obtenus en introduisant en mode ligne :

```
TUTORIAL OPERATIONS=?;
```

L'appel de FSE est l'une des options des menus du domaine IOF. Un exemple en est donné ci-dessous. Le nom de l'environnement apparaît en haut de l'écran.

```
2/5                               environment-name

1 EXEC_PG                          execute a user program
2 EXPAND_FILESET EXPFST             expand a fileset expression
3 FOR77                             F7C  compile FORTRAN 77 program(s)
4 FSE                               call the Full Screen Editor
5 GPL                               compile GPL program(s)
6 LINK_PG                           LK   link an executable module
7 LIST_ACL                          LSACL list access control of objects
8 LIST_CATALOG                      LSCAT list catalog contents
9 LIST_CATSPACE LSCATSP             list catalog extents
10 LIST_DIR                          LSDIR list a directory and its dependents
11 LIST_FILE                         LSF  list file information
12 LIST_FILE_SPACE LSFSP            list extent of file
13 LIST_FILESET                      LSFST list fileset information
14 LIST_GEN                          LSGEN list a generation group
15 LIST_LINK                         LSLK list a file link description
16 LIST_VOLUME                       LSV  list volume
17 LOAD_FILE                         LDF  load a file

-->:_
-:-----
-:-----
+++
```

Généralités et concepts de base

Un 4 doit être introduit dans la zone "-->:" pour obtenir la grille de guidage correspondant à FSE. Cette grille est reproduite ci-dessous.

```
1/1                                FSE                                -->: __
                                     call the Full Screen Editor

LIB                                output library (default is #SLIB)
```

1.4.1.2 Utilisation du mode ligne

Pour appeler FSE en mode ligne, introduire :

```
FSE [LIB = description-bibliothèque];
```

1.4.1.3 Paramètre LIB

Le paramètre LIB spécifie le nom de la bibliothèque SL de sortie dans laquelle vous souhaitez travailler. Si le paramètre LIB est omis, la valeur de la variable système #SLIB (fournie par MWLIB SL) est utilisée. Si #SLIB n'est pas définie (c'est-à-dire si la commande MWLIB SL n'a pas été lancée), la bibliothèque de sortie SL implicite est une bibliothèque temporaire dont le nom est TEMP.SLLIB\$TEMPRY. Cette bibliothèque n'existe que pour la durée de la session IOF, ensuite son contenu est perdu. Pour conserver le contenu de la bibliothèque temporaire, il faut le sauvegarder dans une bibliothèque permanente en utilisant FSE (commande COPY) ou MAINTAIN_LIBRARY, avant la fin de la session.

Pour l'affectation de valeurs à la variable #SLIB et la spécification des bibliothèques d'entrée, se reporter au manuel de référence de l'utilisateur IOF (01 à 04UJ) et à la documentation concernant les processeurs utilisés.

1.4.2 Appel de FSE à partir d'un autre processeur

1.4.2.1 Utilisation des menus

FSE peut être obtenu à partir de plusieurs processeurs, chaque processeur ayant des menus de commandes différents. Le menu reproduit ci-dessous est un exemple de ce qui peut être visualisé pour MAINTAIN_SL_LIBRARY.

```

1/4                                MAINTAIN_SL_LIBRARY

  1 CODE                            encypher member(s) from INLIBi or LIB
  2 COMPARE      CMP  compare member(s) of INLIBi or LIB
  3 COPY         CP   copy member(s) from INLIBi
  4 CREATE                            enter lines in a LIB member
  5 CRLIST                            create a list of members names
  6 DECODE                            decypher member(s) from INLIBi or LIB
  7 DELETE      DL   delete member(s) in LIB
  8 DISPLAY      D   list assigned files, libs or globals
  9 EDIT         ED   call the Text Editor
10 EXEC                            execute LIBMAINT commands from a member
11 FSE                                call the Full Screen Editor
12 INFILE      INF  assign or modify INFILE
13 INLIB1      IL1  assign or modify INLIB1
14 INLIB2      IL2  assign or modify INLIB2
15 INLIB3      IL3  assign or modify INLIB3
16 LIB                            assign or modify LIB
17 LIST        LS   list member(s) name(s)
18 LOWER      LC   convert member(s) to lower case
-->: 11
-----
-:-----
-:-----
+++

```

11 a été introduit dans la zone «-> :» pour appeler FSE. Aucune autre grille de guidage n'apparaîtra car la commande FSE lancée depuis un autre processeur n'a pas de paramètres.

1.4.2.2 Utilisation du mode ligne

Le format de la commande FSE en mode ligne est:

```
FSE;
```

1.5 UTILISATION DE FSE

Après avoir appelé FSE, il est possible de créer et de modifier des unités de bibliothèque. Les unités de bibliothèque sont identifiées par leur nom.

1.5.1 Noms d'unité de bibliothèque

Un nom d'unité de bibliothèque peut avoir 31 caractères au maximum. Les caractères autorisés sont les lettres (A à Z), les chiffres (0 à 9) et les deux caractères spéciaux "souligné" (_) et "moins" (-).

Exemples de noms corrects :

```
UNITE1  
UNI-1  
UNI_1  
99LISTE  
9-TYPE_23
```

1.5.2 Commandes FSE

FSE est composé d'un ensemble de commandes permettant de gérer les bibliothèques origine et de modifier les unités.

Ces commandes sont de deux types :

1. les commandes utilisées pour affecter et libérer les bibliothèques d'entrée et de sortie. Ces commandes sont toujours disponibles lorsque FSE est appelé en tant que processeur indépendant mais ce n'est pas le cas lorsqu'il est appelé à partir de certains processeurs intermédiaires (par exemple, LIBMAINT).
2. les commandes utilisées pour gérer des unités origine. Les unités peuvent être créées et modifiées à l'aide de grilles. Quelle que soit la manière dont FSE a été appelé, ces commandes peuvent toujours être utilisées. A noter qu'il existe des grilles spécifiques pour les différents langages.

FSE - Editeur Plein Ecran

Après appel de FSE, l'un des deux menus ci-dessous s'affiche, le premier dans le cas où FSE a été appelé en tant que processeur indépendant, le second dans le cas où il a été appelé à partir d'un autre processeur.

```
1/5                                F_S_E

  1 COPY          CP  copy member(s) from input to output lib
  2 CREATE        CR  create a member in output lib
  3 DELETE        DL  delete member(s) in output lib
  4 DISPLAY       D   display assigned libraries
  5 INLIB1        IL1 assign or modify INLIB1
  6 INLIB2        IL2 assign or modify INLIB2
  7 INLIB3        IL3 assign or modify INLIB3
  8 LIB           SLLIB assign or modify LIB
  9 LIST          LS  list directory of input or output lib
10 MODIFY        MD  modify member with result in output lib
11 PRINT         PR  display a member of input or output lib
12 QUIT          Q   leave FSE
13 RENAME        RN  rename a member in lib
14 RENUMBER      RB  renumber member(s) in output lib

-->:___
-:-----
-:-----
+++
```

```
1/5                                FSE

  1 COPY          CP  copy member(s) from input to output lib
  2 CREATE        CR  create a member in output lib
  3 DELETE        DL  delete member(s) in output lib
  4 DISPLAY       D   display assigned libraries
  5 LIST          LS  list directory of input or output lib
  6 MODIFY        MD  modify member with result in output lib
  7 PRINT         PR  display a member of input or output lib
  8 QUIT          Q   leave FSE
  9 RENAME        RN  rename a member in lib
10 RENUMBER      RB  renumber member(s) in output lib

-->:___
-:-----
-:-----
+++
```

Généralités et concepts de base

Ces deux menus donnent la liste des commandes disponibles pour un FSE indépendant (F_S_E) et pour un FSE appelé depuis un autre processeur. La seule différence se situe au niveau des commandes LIB, INLIB1, INLIB2 et INLIB3 qui ne sont disponibles que pour le FSE indépendant.

Dans le cas où FSE est indépendant, il est possible d'affecter et de libérer les bibliothèques au moment où FSE est lancé ou une fois dans le domaine FSE.

Dans le cas où FSE est appelé à partir d'un autre processeur, par contre, les bibliothèques origine doivent être affectées avant de lancer FSE. Cela signifie qu'il n'est pas possible d'affecter ou de libérer les bibliothèques une fois dans le domaine FSE.

Toutes les commandes FSE sont décrites en détail au chapitre 2.

1.6 GRILLES PROPRES AUX DIFFERENTS LANGAGES

FSE fournit des grilles standards destinées à guider l'utilisateur qui veut créer du code origine et ce, dans tous les langages standards utilisés sous GCOS 7. Avec ces grilles, les différentes parties des articles de données à introduire ou à modifier sont visualisées à l'écran.

2. Description des commandes

Ce chapitre contient la description détaillée de chaque commande FSE avec exemples d'utilisation. Les commandes, classées dans l'ordre alphabétique pour faciliter la consultation, sont les suivantes :

- **COPY** (*abréviation CP*)
- **CREATE** (*abréviation CR*)
- **DELETE** (*abréviation DL*)
- **DISPLAY** (*abréviation D ou DISP*)
- **INLIB1** (*abréviation IL1*)
- **INLIB2** (*abréviation IL2*)
- **INLIB3** (*abréviation IL3*)
- **LIB** (*pas d'abréviation*)
- **LIST** (*abréviation LS*)
- **MODIFY** (*abréviation MD*)
- **PRINT** (*abréviation PR*)
- **QUIT** (*abréviation Q*)
- **RENAME** (*abréviation RN*)
- **RENUMBER** (*abréviation RB*)

La description de chaque commande commence sur une page nouvelle. Les renseignements fournis pour chacune des commandes sont les suivants :

- fonction de la commande
- format de la grille de guidage
- format de la commande en mode ligne
- description des paramètres
- valeurs implicites (éventuelles)
- si nécessaire, description de l'exécution de la commande
- exemples, en mode ligne ou en mode écran

2.1 COMMANDE COPY

La commande COPY (abréviation CP) permet de copier une ou plusieurs unités d'une bibliothèque d'entrée dans la bibliothèque de sortie.

Grille de guidage

```

1/1                                COPY                                -->:  __
                                copy member(s) from input or output lib

MEMBERS                          + members to be processed
                                *
                                -----
INLIB                             + INLIB1,INLIB2,INLIB3           INLIB1
                                -----
REPLACE                          do you want to overwrite ?       0
                                -----
BRIEF                             brief report ?                   0
                                -----
-----

```

Format en mode ligne

```

COPY
[ MEMBERS = { nom-unité
              { nom-astérisque }
              { *
              { _
            ]

[ INLIB = { INLIB1
           { INLIB2
           { INLIB3
        ]

[ REPLACE ]

[ BRIEF ];

```

Description des commandes

Paramètres

Le paramètre MEMBERS spécifie la ou les unités à copier. Il est possible d'introduire un nom d'unité ou bien d'utiliser la convention astérisque pour désigner un ensemble d'unités (la valeur implicite est un astérisque, ce qui correspond à toutes les unités de la bibliothèque indiquée). A l'exécution de la commande, il y a visualisation d'informations concernant l'unité ou les unités spécifiées et d'une zone OMIT pour chaque unité. Si on ne veut pas copier une ou plusieurs unités, remplir les zones OMIT correspondantes. N'importe quel caractère peut être introduit dans la zone OMIT, à l'exception du point d'interrogation (?) qui appelle le texte HELP. Les unités dont la zone OMIT est laissée en blanc seront copiées.

Le paramètre INLIB spécifie l'une des bibliothèques d'entrée INLIB1, INLIB2 ou INLIB3. Si ce paramètre est omis, la valeur implicite est INLIB1.

Le paramètre REPLACE permet de remplacer une unité existant déjà dans la bibliothèque de sortie par l'unité copiée portant le même nom.

Le paramètre BRIEF demande que seul le nom des unités soit visualisé; les informations comme la date et l'heure de la dernière modification, le langage et le nombre de lignes ne seront pas visualisées.

Remarque : La convention astérisque est utilisée dans un profil de nom pour représenter n'importe quel caractère (y compris zéro caractère).

Exemples :

SP_*	désigne toutes les unités dont le nom commence par "SP_".
*_V3	désigne toutes les unités dont le nom finit par "_V3".
*	désigne toutes les unités de la bibliothèque.

Exécution de la commande

Une grille de guidage spécifique de la commande COPY est visualisée lorsque les paramètres de la commande ont été acceptés. Cette grille comporte la liste des unités de la bibliothèque d'entrée spécifiée susceptibles d'être copiées. Les en-têtes sont :

OMIT	Pour que l'unité ne soit pas copiée, introduire n'importe quel caractère, à l'exception du point d'interrogation (?).
NAMES	Nom de l'unité
SAVED	Date et heure de la dernière modification, sous la forme: mm/jj/aa hh:mm
LANG	Langage de l'unité
LINES	Nombre de lignes de l'unité

FSE - Editeur Plein Ecran

Quand les zones OMIT des unités ne devant pas être copiées ont été remplies, appuyer sur la touche de transmission (TRANS). Une liste apparaît sur l'écran indiquant les unités copiées et celles qui ne l'ont pas été. La grille suivante (s'il y en a une) des unités susceptibles d'être copiées est alors visualisée.

Pour mettre fin à l'exécution de la commande COPY, introduire une barre oblique (/) dans la zone "-->:"; les unités dont les noms sont visualisés ne seront pas copiés. Sur la page suivante figure un exemple d'utilisation de grille pendant l'exécution d'une commande COPY.

```

FSE                                -->:___                                COPY

      INPUT LIBRARY = LINT.CMD.SLLIB
      OUTPUT LIBRARY = :000137.TEMP.SLLIB

OMIT---NAMES---                    ----SAVED-----    -LANG-    -LINES-
N   H_ADL_ECOMPL                    01/28/84 10:58    GPL        219
N   H_ADL_EDIR                      03/26/83 14:28    GPL         59
      H_ADL_ELOAD                    03/26/83 14:29    GPL         27
      H_ADL_EMAIN                    03/26/83 14:30    GPL         25
N   H_ADL_EPRISF                    03/26/83 14:30    GPL         27
      H_ADL_ESAVE                    10/18/83 17:05    GPL        234
N   H_ADL_ETEST                     11/19/83 17:41    GPL         50
      H_ADL_EUNSVE                   03/26/83 14:30    GPL         18
N   H_ADL_EXANAL                    10/13/83 13:52    GPL         67
      H_ADL_MGPL                     10/13/83 15:36    ADL         20
N   H_ADL_TOKGEN                    06/09/83 12:09    GPL         79
      H_BIN_A_CCBX                   04/02/83 18:45    ADL         13
N   H_BIN_A_CCOB                    04/02/83 18:46    ADL         13
N   H_BIN_A_CDAT                    12/06/83 18:27    DAT         28
-----

```

Dans l'exemple ci-dessus, la commande COPY H_* a été émise. La grille ci-dessus est visualisée. Ensuite la lettre N est introduite dans la zone OMIT de chaque unité à ne pas copier. Cette opération est suivie d'un appui sur la touche de transmission (TRANS). Le résultat apparaît dans la grille suivante.

```

FSE                                -->:___                                COPY

      INPUT LIBRARY = LINT.CMD.SLLIB
      OUTPUT LIBRARY = :000137.TEMP.SLLIB

OMIT---NAMES---                    ----SAVED-----    -LANG-    -LINES-
N   H_ADL_ECOMPL                    NOT COPIED
N   H_ADL_EDIR                      NOT COPIED
      H_ADL_ELOAD                    COPIED
      H_ADL_EMAIN                    COPIED
N   H_ADL_EPRISF                    NOT COPIED
      H_ADL_ESAVE                    COPIED
N   H_ADL_ETEST                     NOT COPIED
      H_ADL_EUNSVE                   COPIED
N   H_ADL_EXANAL                    NOT COPIED
      H_ADL_MGPL                     COPIED
N   H_ADL_TOKGEN                    NOT COPIED
      H_BIN_A_CCBX                   COPIED
N   H_BIN_A_CCOB                    NOT COPIED
N   H_BIN_A_CDAT                    NOT COPIED
-----

```

2.2 COMMANDE CREATE

La commande CREATE (abréviation CR) permet :

- de créer une nouvelle unité dans la bibliothèque de sortie,
- d'ajouter des données à la fin d'une unité existante,
- de remplacer une unité existante.

Pour chaque langage, le système fournit une application standard qui permet de créer une unité en mode formaté.

Grille de guidage

```

1/1                                CREATE                                -->:  __
                                create a member in output lib

MEMBER                            + member to be processed

                                -----
LANG                               language type                                DAT
                                -----
REPLACE                           do you want to overwrite ?                0
                                -----
APPEND                             append in member                          0
                                -----
-----

```

Format en mode ligne

```

CREATE
    MEMBER = nom-unité
    [LANG = langage]
    [REPLACE]
    [APPEND]

```

Paramètres

Le paramètre MEMBER indique le nom de l'unité.

Le paramètre LANG spécifie le langage associé à l'unité à créer ou à ajouter à une unité existante. Si ce paramètre est omis, la valeur implicite est DAT.

Le paramètre APPEND indique que les données sont à ajouter à la fin d'une unité existante.

Le paramètre REPLACE permet de recouvrir dans la bibliothèque de sortie une unité existante portant le même nom. Le paramètre REPLACE ne peut pas être utilisé avec le paramètre APPEND.

Exécution de la commande

Lorsque les informations nécessaires ont été introduites, l'unité est créée ou ajoutée à une unité existante dans la bibliothèque de sortie. Un guidage est affiché pour chaque ligne de texte à introduire. Une fois introduites, ces lignes constitueront l'unité de bibliothèque.

Les formats dans les différents langages sont les formats standards :

Langage	Format (numéro des colonnes)
COBOL (COB)	1-6 : numéro de ligne, 7-80 : texte
COBOLX (CBX)	1-6 : numéro de ligne, 7-72 : texte
DATASSF (DAT)	pas de numéro de ligne, longueur d'article=longueur du texte (255 caractères maxi)
FORTRAN (FOR)	pas de numéro de ligne, 1-72 : texte
GPL	1-6 : numéro de ligne, 7-80 : texte
JCL	pas de numéro de ligne, longueur d'article=longueur du texte (255 caractères maxi)
CMD	pas de numéro de ligne, longueur d'article=longueur du texte (255 caractères maxi)
PASCAL (PAS)	pas de numéro de ligne, longueur d'article=longueur du texte (255 caractères maxi)
C_LANG (CL)	pas de numéro de ligne, longueur d'article=longueur du texte (255 caractères maxi)

Si une ligne plus longue que ne l'autorise le format standard correspondant est introduite, elle sera tronquée.

Pour mettre fin à la session CREATE, introduire une barre oblique (/) dans la zone EXIT de la grille de saisie.

La signification de la zone NEXT dépend de l'application (fournie par le système) correspondant au langage. Pour plus de détails, se reporter aux textes d'aide (helps) existant pour chaque application.

Description des commandes

Exemples :

```
CR jh_mem LANG=PAS REPLACE;
```

Il y a passage en mode saisie pour créer (ou remplacer si elle existe déjà) une unité appelée jh_mem dans la bibliothèque LIB.

```
CR my_program LANG=CBX APPEND;
```

Il y a passage en mode saisie pour ajouter des lignes de texte à une unité existante appelée "my_program" dans la bibliothèque LIB. Le langage est le COBOLX. La page qui suit montre les grilles utilisées pour cet exemple.

```
H_FSE_CCBX REC CBX EXIT__NEXT__MY-PROGRAM CREATE
```

```
10 IDENTIFICATION DIVISION.
```

```
COL 8-72:  
PROGRAM-ID. MYPROGRAM.
```

```
-----  
COL 7: _____  
-----
```

L'article 10 était déjà saisi. L'article suivant a été saisi dans les colonnes 8 à 72 (la colonne 7 a été laissée en blanc). Noter que pour avoir des "espaces" dans les colonnes 8 à 11, il suffit d'introduire un espace dans la première colonne de cette zone. Pour chaque langage une explication détaillée est donnée dans le texte HELP. Pour l'obtenir introduire un point d'interrogation (?) dans la zone EXIT. Après avoir appuyé sur la touche de transmission (TRANS), la grille suivante s'affiche :

```
H_FSE_CCBX REC CBX EXIT__NEXT__MY-PROGRAM CREATE
```

```
10 IDENTIFICATION DIVISION.
```

```
20 PROGRAM-ID. MYPROGRAM.
```

```
COL 8-72:
```

```
COL 7: _____  
-----
```

2.3 COMMANDE DELETE

La commande DELETE (abréviation DL) permet de supprimer une ou plusieurs unités de la bibliothèque de sortie.

Grille de guidage

```

1/1                                DELETE                                -->:  __
                                delete member(s) in output lib

MEMBERS                            + candidates for processing
                                -----
BRIEF                              brief report ?                                0
                                -
-----

```

Format en mode ligne

```

DELETE

[MEMBERS = { nom-unité
             { nom-astérisque }
             *
             -
]
[BRIEF];

```

Paramètres

Le paramètre MEMBERS spécifie la ou les unités à supprimer. Il est possible d'introduire le nom d'une seule unité ou bien d'utiliser la convention astérisque pour désigner un ensemble d'unités. Les informations concernant la ou les unités spécifiées sont alors visualisées avec une zone OMIT pour chaque unité. Pour ne pas supprimer une ou plusieurs unités, remplir les zones OMIT correspondantes. N'importe quel caractère peut être introduit à cet effet dans la zone OMIT, à l'exception du point d'interrogation (?) qui appelle le texte HELP. Les unités dont la zone OMIT est laissée en blanc seront supprimées.

Le paramètre BRIEF demande que seul le nom des unités soit visualisé; les informations telles que la date et l'heure de la dernière modification, le langage et le nombre de lignes ne le seront pas.

Description des commandes

Exécution de la commande

Une grille spécifique de la commande DELETE est visualisée lorsque les paramètres de la commande ont été acceptés. Cette grille contient la liste des unités de la bibliothèque de sortie susceptibles d'être supprimées. Les en-têtes sont :

OMIT	Pour que l'unité ne soit pas supprimée, introduire n'importe quel caractère à l'exception de ?
NAMES	Nom de l'unité
SAVED	Date et heure de la dernière modification, sous la forme mm/jj/aa hh:mm
LANG	Langage de l'unité
LINES	Nombre de lignes de l'unité

Quand les zones OMIT des unités ne devant pas être supprimées ont été remplies, appuyer sur la touche de transmission (TRANS). Une liste apparaît sur l'écran indiquant les unités supprimées et celles qui ne l'ont pas été. La grille suivante (s'il y en a une) des unités susceptibles d'être supprimées est alors visualisée.

Pour mettre fin à l'exécution de la commande DELETE, introduire une barre oblique (/) dans la zone " --> "; les unités dont les noms sont visualisés ne seront pas supprimées.

Exemples :

```
DL  jh_mem;
```

L'unité jh_mem dans LIB est supprimée.

```
DL  H_*;
```

Toutes les unités commençant par H_ sont supprimées. L'utilisation de la grille pour cet exemple est montré à la page qui suit.

```
FSE                                -->: __                                DELETE

      LIBRARY = TEMP.SLLIB$TEMPRY

OMIT---NAMES---                ----SAVED-----  -LANG-  -LINES-
N   H_ADL_ECOMPL                01/28/84 10:58      GPL      2192
N   H_ADL_EDIR                  03/26/83 14:28      GPL        599
N   H_ADL_ELOAD                 03/26/83 14:29      GPL        270
N   H_ADL_EMAIN                 03/26/83 14:30      GPL        251
N   H_ADL_EPRISF                03/26/83 14:30      GPL        274
N   H_ADL_ESAVE                 10/18/83 17:05      GPL      2349
N   H_ADL_ETEST                 11/19/83 17:41      GPL        500
N   H_ADL_EUNSVE                03/26/83 14:30      GPL        180
N   H_ADL_EXANAL                10/13/83 18:52      GPL        674
N   H_ADL_MGPL                  10/13/83 15:36      GPL        209
      H_ADL_TOKGEN                06/09/83 12:09      GPL        791
      H_BIN_A_CCBX                 04/02/83 18:45      ADL        134
      H_BIN_A_CCOB                 04/02/83 18:46      ADL        138
      H_BIN_A_CDAT                 12/06/83 18:27      DAT        283
-----
```

Pour cet exemple, la commande qui a été émise est DL H_*. La grille ci-dessus est alors visualisée. La lettre N est introduite dans la zone OMIT pour chacune des unités à ne pas supprimer. Appuyer ensuite sur la touche de transmission (TRANS). Le résultat s'affiche sur la grille suivante.

```
FSE                                -->: __                                DELETE

      LIBRARY = TEMP.SLLIB$TEMPRY

OMIT---NAMES---                ----SAVED-----  -LANG-  -LINES-
--   H_ADL_ECOMPL                NOT DELETED
--   H_ADL_EDIR                  NOT DELETED
--   H_ADL_ELOAD                 NOT DELETED
--   H_ADL_EMAIN                 NOT DELETED
--   H_ADL_EPRISF                NOT DELETED
--   H_ADL_ESAVE                 NOT DELETED
--   H_ADL_ETEST                 NOT DELETED
--   H_ADL_EUNSVE                NOT DELETED
--   H_ADL_EXANAL                NOT DELETED
--   H_ADL_MGPL                  NOT DELETED
--   H_ADL_TOKGEN                DELETED
--   H_BIN_A_CCBX                 DELETED
--   H_BIN_A_CCOB                 DELETED
--   H_BIN_A_CDAT                 DELETED
-----
```

2.4 COMMANDE DISPLAY

La commande DISPLAY (abréviation D ou DISP) permet de lister les bibliothèques affectées.

Grille de guidage

```

1/1                                DISPLAY                                -->:  __
                                display assigned libraries

OPTION + LIBS, LIB, INLIB1, INLIB2 or INLIB3                                LIBS
_____
-----

```

Format en mode ligne

```

DISPLAY

[OPTION = { LIB
           { INLIBn } } ] ;
           { LIBS
           ---

```

Paramètres

Le paramètre OPTION spécifie les noms symboliques des bibliothèques à lister. Introduire LIB, INLIB1, INLIB2 ou INLIB3 pour lister une bibliothèque déterminée (c'est-à-dire pour connaître le nom de la bibliothèque affectée correspondante et le support sur lequel elle est implantée), ou LIBS pour lister toutes les bibliothèques affectées.

Si la bibliothèque demandée n'est pas affectée, le message LIBRARY NOT ASSIGNED est visualisé. De même, si LIBS est introduit et qu'aucune bibliothèque n'est affectée, c'est le message NO ACTIVE LIBRARY qui est visualisé.

Si le paramètre OPTION est omis, la valeur implicite est LIBS.

Exemple :

```
D;
```

Toutes les bibliothèques affectées sont visualisées.

2.5 COMMANDE INLIBn

Les commandes INLIB1 (abréviation IL1), INLIB2 (abréviation IL2) et INLIB3 (abréviation IL3) permettent d'affecter des bibliothèques d'entrée avec les noms symboliques INLIB1, INLIB2 et INLIB3 respectivement.

Ces commandes n'ont d'effet que pendant l'exécution de FSE.

Grille de guidage

```
1/1                               INLIB1                               -->:  __
                                assign or modify INLIB1
```

New identification of INLIB1. No name cancels INLIB1.
LIBRARY

Format en mode ligne

```
INLIBn
      [LIBRARY = description-bibliothèque] ;
```

Paramètre

Le paramètre LIBRARY spécifie la bibliothèque à affecter. Si une bibliothèque d'entrée portant le même nom symbolique a déjà été affectée (soit par l'une des commandes INLIB1, INLIB2 ou INLIB3, soit par la commande GCL MWINLIB), elle sera remplacée par la bibliothèque spécifiée. Si aucune description de bibliothèque n'est fournie, la bibliothèque d'entrée correspondante cesse d'être affectée.

Format des descriptions de bibliothèque :

Une description de bibliothèque peut avoir l'un des trois formats suivants, selon qu'il s'agit d'une bibliothèque cataloguée, d'une bibliothèque non cataloguée ou d'une bibliothèque temporaire :

- Bibliothèque cataloguée :

nom-d'accès-bibliothèque [/genver] [\$CAT[n]]

- Bibliothèque non cataloguée :

nom-d'accès-bibliothèque [:support:classe-app] [\$UNCAT] [\$RES]

- Bibliothèque temporaire :

nom-d'accès-bibliothèque [:support:classe-app] \$TEMPRY

Signification des paramètres :

nom-accès-bibliothèque

genver	nom d'accès de la bibliothèque (ex.DEPT.SERVICE.LIB)
\$CAT[n]	numéro de génération et/ou de version (ex. G0013V11) indique une bibliothèque cataloguée. Si n est spécifié, il indique le numéro (rang) du catalogue privé (1 <= n <= 5)
\$UNCAT	indique une bibliothèque non cataloguée (c'est la valeur implicite si vous spécifiez un support et une classe d'appareils)
\$RES	indique une bibliothèque résidante (c'est la valeur implicite pour une bibliothèque temporaire). Avec \$RES, ni le support, ni la classe d'appareils ne doivent être spécifiés.
\$TEMPRY	indique une bibliothèque temporaire

Exemples :

INLIB3

La bibliothèque portant actuellement le nom symbolique INLIB3 cesse d'être affectée.

INLIB1 DEPT.LIB1

Affectation de la bibliothèque cataloguée DEPT.LIB1 à INLIB1.

INLIB2 P1.DD\$CAT

Affectation de la bibliothèque cataloguée P1.DD à INLIB2.

INLIB1 P.L/G0005

Affectation de la génération 5 de la bibliothèque cataloguée P.L à INLIB1.

INLIB3 LIBUNC:K181:MS/M452

Affectation de la bibliothèque non cataloguée LIBUNC à INLIB3.

INLIB1 LIBRES\$RES

Affectation de la bibliothèque résidante non cataloguée LIBRES à INLIB1.

INLIB2 DTEMP\$TEMPRY

Affectation de la bibliothèque temporaire DTEMP à INLIB2.

INLIB1 WK:K753:MS/M452\$TEMPRY

Affectation de la bibliothèque temporaire WK (implantée dans le volume privé K753) à INLIB1.

2.6 COMMANDE LIB

La commande LIB (pseudonyme SLLIB) permet d'affecter une bibliothèque de sortie avec le nom symbolique LIB.

Cette commande n'a d'effet que pendant l'exécution de FSE.

Grille de guidage

```
1/1                                LIB                                -->:  __
                                assign or modify LIB
```

```
New identification of LIB. No name cancels LIB.
LIBRARY
```

Format en mode ligne

```
LIB [LIBRARY = description-bibliothèque] ;
```

Paramètre

Le paramètre LIBRARY spécifie la bibliothèque de sortie à affecter. Si une bibliothèque de sortie a déjà été affectée (soit par la commande LIB, soit par la commande GCL MWLIB), elle est remplacée par la bibliothèque spécifiée. Si aucune description de bibliothèque n'est fournie, la bibliothèque de sortie actuellement affectée cesse de l'être.

Format des descriptions de bibliothèque

Une description de bibliothèque peut avoir l'un des trois formats suivants, selon qu'il s'agit d'une bibliothèque cataloguée, d'une bibliothèque non cataloguée ou d'une bibliothèque temporaire :

- Bibliothèque cataloguée
nom-d'accès-bibliothèque [/genver] [\$CAT[n]]
- Bibliothèque non cataloguée
nom-d'accès-bibliothèque [:support:classe-app] [\$UNCAT] [\$RES]
- Bibliothèque temporaire
nom-d'accès-bibliothèque [:support:class-app] \$TEMPRY

Signification des paramètres

nom-accès-bibliothèque

genver	nom d'accès de la bibliothèque (ex. DEPT.SERVICE.LIB)
\$CAT[n]	numéro de génération et/ou de version (ex. G0013V11) indique une bibliothèque cataloguée. Si n est spécifié, il indique le numéro (rang) du catalogue privé (1 <= n <= 5)
\$UNCAT	indique une bibliothèque non cataloguée (c'est la valeur implicite si le support et la classe d'appareils sont spécifiés)
\$RES	indique une bibliothèque résidante (c'est la valeur implicite pour une bibliothèque temporaire). Avec \$RES, ni le support, ni la classe d'appareils ne doivent être spécifiés.
\$TEMPRY	indique une bibliothèque temporaire.

Exemples

LIB

La bibliothèque de sortie actuellement affectée cesse de l'être.

LIB DEPT.LIB1

Affectation de la bibliothèque cataloguée DEPT.LIB1 en tant que bibliothèque de sortie courante.

LIB P1.DD\$CAT

Affectation de la bibliothèque cataloguée P1.DD en tant que bibliothèque de sortie courante.

LIB P.L/G0005

Affectation de la génération 5 de la bibliothèque cataloguée P.L en tant que bibliothèque de sortie courante.

LIB LIBUNC:K181:MS/M452

Affectation de la bibliothèque non cataloguée LIBUNC en tant que bibliothèque de sortie courante.

LIB LIBRES\$RES

Affectation de la bibliothèque résidante non cataloguée LIBRES en tant que bibliothèque de sortie courante.

LIB DTEMP\$TEMPRY

Affectation de la bibliothèque temporaire DTEMP en tant que bibliothèque de sortie courante.

LIB WK:K753:MS/M452\$TEMPRY

Affectation de la bibliothèque temporaire WK (implantée dans le volume privé K753) en tant que bibliothèque de sortie courante.

2.7 COMMANDE LIST

La commande LIST (abréviation LS) permet d'obtenir des informations concernant les unités d'une bibliothèque spécifiée.

Ces informations comprennent pour chaque unité la date et l'heure de la dernière modification, le langage et le nombre de lignes.

Grille de guidage

```

1/1                                LIST                                -->:  __
                                list directory of input or output lib

MEMBERS                          + member(s) to be processed
                                *
                                -----
LIBRARY                          + LIB, INLIB1, INLIB2, INLIB3      LIB
                                -----
BRIEF                            brief report ?                      0
                                -
-----

```

Format en mode ligne

```

LIST

[MEMBERS = { nom-unité
            { nom-astérisque }
            { *
            -
            } ]

[LIBRARY = { INLIB1
            { INLIB2 }
            { INLIB3 }
            { LIB
            ----
            } ]

[BRIEF] ;

```

Paramètres

Le paramètre MEMBERS spécifie la ou les unités sur lesquelles vous souhaitez avoir des informations. Introduire un nom d'unité ou utiliser la convention astérisque pour désigner un ensemble d'unités (la valeur implicite étant l'astérisque seul, ce qui correspond à toutes les unités de la bibliothèque spécifiée).

Le paramètre LIBRARY spécifie le nom de la bibliothèque: LIB, INLIB1, INLIB2 ou INLIB3. Si ce paramètre est omis, la valeur implicite est LIB.

Le paramètre BRIEF demande que seul le nom des unités soit visualisé. La date et l'heure de la dernière modification de chaque unité, le langage et le nombre de lignes n'apparaîtront pas.

Exécution de la commande

A l'exécution, une grille différente est visualisée selon que le paramètre BRIEF est ou non spécifié (voir exemples ci-dessous).

Cette grille contient le nom, et les informations complémentaires éventuellement, des unités de la bibliothèque spécifiée. Les en-têtes sont les suivants :

NAMES	nom de l'unité
SAVED	date et heure de la dernière modification, sous la forme mm/jj/aa hh:mm
LANG	langage
LINES	nombre de lignes de l'unité

L'identification de la bibliothèque ainsi que le nombre de blocs et le pourcentage d'espace utilisé dans cette bibliothèque sont également visualisés.

Les informations détaillées fournies pour les unités listées peuvent exiger plusieurs grilles. Pour passer d'une grille à la suivante, introduire un caractère "supérieur à" (>) ou un espace dans la zone "-->:". Pour rappeler la grille précédente, introduire un caractère "inférieur à" (<) dans la zone ">:".

La dernière grille indique le nombre d'unités listées et le nombre total de lignes (s'il s'agit d'un listage détaillé).

Si un compte rendu non détaillé (paramètre BRIEF) a été demandé, deux colonnes de noms sont visualisées.

Pour mettre fin à l'exécution de la commande LIST, introduire une barre oblique (/) dans la zone "-->:".

Exemples :

```
LS jh_mem* ;
```

Toutes les unités de la bibliothèque LIB commençant par jh_mem sont listées.

```
LS jh_mem* LIBRARY=INLIB2 ;
```

Toutes les unités de la bibliothèque INLIB2 commençant par jh_mem sont listées.

```
LS H_* ;
```

Description des commandes

Toutes les unités de la bibliothèque LIB commençant par H_ sont listées avec des informations détaillées. La grille suivante s'affiche :

```

FSE                                -->:___                                LIST

      LIBRARY = LINT.CMB.SLLIB      CATALOGUED
      USED BLOCKS = 1815/3118      (58%)

---NAMES---                        ----SAVED-----      -LANG-      -LINES-
H_ADL_ECOMPL                       01/28/84 10:58      GPL          2192
H_ADL_EDIR                          03/26/83 14:28      GPL           599
H_ADL_ELOAD                          03/26/83 14:29      GPL           270
H_ADL_EMAIN                          03/26/83 14:30      GPL           251
H_ADL_EPRISF                         03/26/83 14:30      GPL           274
H_ADL_ESAVE                          10/18/83 17:05      GPL          2349
H_ADL_ETEST                          11/19/83 17:41      GPL           500
H_ADL_EUNSVE                         03/26/83 14:30      GPL           180
H_ADL_EXANAL                         10/13/83 18:52      GPL           674
H_ADL_MGPL                          02/13/83 15:36      ADL           209
H_ADL_TOKGEN                         06/09/83 12:09      GPL           791
H_BIN_A_CCBX                        04/02/83 18:45      ADL           134
H_BIN_A_CCOB                        04/02/83 18:46      ADL           138
H_BIN_A_CDAT                        12/06/83 18:27      DAT           283
-----
      LS H_* BRIEF;
  
```

Toutes les unités de la bibliothèque LIB commençant par H_ sont listées sans aucune information détaillée. La grille suivante s'affiche :

```

FSE                                -->:___                                LIST

      LIBRARY = LINT.CMB.SLLIB      CATALOGUED
      USED BLOCKS = 1815/3118      (58%)

---NAMES---                        ---NAMES---
ACTFOR_A_CONSERVER                  H_ADL_ECOMPL
ADL                                  H_ADL_EDIR
BID_DSYNTAX                          H_ADL_ELOAD
BID_TOTO                              H_ADL_EMAIN
BIDOU_BATCH                          H_ADL_EPRISF
BIDOU_GPL                             H_ADL_ESAVE
BIDOU_LINK                           H_ADL_ETEST
BIDOU_PUT                             H_ADL_EUNSVE
BIDOUBATCH                           H_ADL_EXANAL
DEMO_FSE                              H_ADL_MGPL
DEMO_FSE2                             H_ADL_TOKGEN
FRM_ATTRL                            H_BIN_A_CCBX
FRM_ATTRL_A_CONSERVER                H_BIN_A_CCOB
FSE_SHOW                              H_BIN_A_CDAT
-----
  
```

2.8 COMMANDE MODIFY

La commande MODIFY (abréviation MD) permet de modifier le contenu d'une unité dans la bibliothèque de sortie (LIB), ou de créer une nouvelle unité. Les modifications s'effectuent au moyen de demandes. Pour la description détaillée de la session MODIFY, se reporter au chapitre 3. Celle des demandes en mode ligne figure dans le chapitre 4. Quant aux fonctions de modification en mode écran, on les trouvera au chapitre 5.

Grille de guidage

```

1/1                                MODIFY                                -->:  __
                                modify member with result in output lib

MEMBER                            member to be processed
-----
LIBRARY                            LIB, INLIB1, INLIB2, INLIB3      LIB
-----
NEW                                new name of the member
-----
REPLACE                            do you want to overwrite?
-----
LANG                                language type
-----
-----

```

Format en mode ligne

```

MODIFY

[MEMBER    = nom-unité]

[LIBRARY = {INLIB1}
           {INLIB2}]
           {INLIB3}
           {LIB
           ---}

[NEW = nom-unité]
[REPLACE]
[LANG = langage];

```

Paramètres

Le paramètre MEMBER spécifie l'unité à créer ou à modifier.

Le paramètre LIBRARY spécifie dans quelle bibliothèque (LIB, INLIB1, INLIB2 ou INLIB3) l'unité à modifier doit être lue (la valeur implicite est LIB).

Le paramètre NEW fournit le nouveau nom d'une unité si celui-ci doit être changé.

Le paramètre REPLACE précise que la nouvelle unité doit remplacer dans la bibliothèque de sortie (LIB) une ancienne unité ayant éventuellement le même nom. Il n'y a pas lieu de spécifier ce paramètre lorsqu'il s'agit de modifier une unité sans changer son nom.

Le paramètre LANG (pseudonyme TYPE) précise le langage de la nouvelle unité. La valeur implicite est le langage de l'unité spécifiée par le paramètre MEMBER. Si le paramètre MEMBER est omis, la valeur implicite est DAT.

Si le paramètre MEMBER n'est pas spécifié, aucune unité ne sera chargée dans le tampon principal et aucun nom implicite ne sera utilisé, sauf s'il y a un paramètre NEW. Le tampon devra être chargé en utilisant une des demandes de lecture, d'adjonction ou d'insertion de texte. Le nom de l'unité peut aussi être spécifié par la demande d'attribution de nom ou de sauvegarde.

Exécution de la commande

Lorsque les informations nécessaires ont été introduites, l'unité, si elle est spécifiée, est chargée dans le tampon. La grille du mode ligne est alors visualisée.

L'unité peut faire l'objet de n'importe laquelle des demandes de modification ; il est également possible, en remplissant les zones appropriées, de renuméroter les lignes de l'unité, de la sauvegarder ou de repasser au niveau commande. Un texte HELP est disponible pour chaque demande de modification.

Toutes les modifications sont exécutées dans le tampon. Tant que le contenu de ce tampon n'a pas été explicitement sauvegardé, l'unité modifiée n'est pas intégrée à la bibliothèque de sortie.

Deux modes sont utilisables avec la commande MODIFY :

- le mode ligne (lancé par la demande LE)
- le mode écran (lancé par la demande SE)

En mode ligne, les demandes doivent obligatoirement porter sur des lignes spécifiées. En mode écran, les modifications peuvent être appliquées localement à la totalité de la grille courante mais il est également possible d'utiliser des demandes "ligne" comme en mode ligne.

2.9 COMMANDE PRINT

La commande PRINT (abréviation PR) permet de visualiser le contenu d'une unité spécifiée.

Grille de guidage

```

1/1                                PRINT                                -->: __
                                print a member of input or output lib

MEMBER                            + member to be processed
                                -----
LIBRARY                            + LIB, INLIB1, INLIB2, INLIB3      LIB
                                -----
WINDOWING                          0 ---> fold, 1 ---> windowing      0
                                -
LNUMBER                             default line numbering ?      0
                                -
START                               number of the first line to print  -
                                -----
                                -----

```

Format en mode ligne

```

PRINT

MEMBER = nom-unité

[LIBRARY = { INLIB1 }
           { INLIB2 } ]
           { INLIB3 }
           { LIB   }
           ---

[START = numéro_ligne]

[WINDOWING]

[NUMBER];

```

Description des commandes

Paramètres

Le paramètre MEMBER spécifie l'unité à traiter.

Le paramètre LIBRARY indique dans quelle bibliothèque l'unité se trouve : LIB, INLIB1, INLIB2 ou INLIB3. Si ce paramètre est omis, la valeur implicite sera LIB.

Le paramètre START spécifie le numéro de ligne du premier article à lire lorsque vous ne désirez pas imprimer une unité à partir du commencement. Tous les articles qui suivent seront lus et affichés, grille par grille, jusqu'à ce que le caractère "barre oblique" (/) soit introduit dans la zone de gestion (ctl).

Le paramètre WINDOWING demande un déplacement latéral de la fenêtre. Il n'y a pas de pliage de lignes; pour visualiser les caractères qui ne peuvent tenir tous sur une seule ligne de l'écran, introduire L ou R dans la zone "-->:". La fenêtre est déplacée de 30 caractères sur la gauche ou sur la droite respectivement.

Le paramètre LNUMBER spécifie que pour les unités dont le format de langage ne prévoit pas de numéro de ligne (comme avec DAT, par exemple), les numéros de lignes doivent être affichés à l'écran avec l'unité de bibliothèque. Ce paramètre n'est pas pris en compte pour les unités dont le format comporte un numéro de ligne (par exemple COB).

Exécution de la commande

Une grille de guidage spécifique de la commande PRINT est visualisée, selon la valeur donnée au paramètre WINDOWING (voir exemples ci-dessous) .

Une fenêtre montrant le contenu de l'unité spécifiée de la bibliothèque désignée apparaît sur l'écran. Les caractères suivants peuvent être introduits dans la zone "-->:" :

Caractère Fonction

> (supérieur à) ou espace	déplacement de la fenêtre d'un écran vers le bas
< (inférieur à)	déplacement de la fenêtre d'un écran vers le haut
D	déplacement de la fenêtre d'un demi-écran vers le bas
U	déplacement de la fenêtre d'un demi-écran vers le haut
R	déplacement de la fenêtre de 30 caractères sur la droite (*)
L	déplacement de la fenêtre de 30 caractères sur la gauche (*)
/	arrêt de l'exécution de la commande et retour au niveau commande.

(*) ces options ne sont disponibles que si le mode WINDOWING a été demandé.

Exemples :

PR H_ADL_ACTFOR;

L'unité H_ADL_ACTFOR dans la bibliothèque LIB est visualisée sans les fonctions WINDOWING. Voir la grille ci-dessous.

```

FSE                                -->:___                                PRINT

      LIBRARY = LINT.CMB.SLLIB
      MEMBER = H_ADL_ACTFOR
-----
10 H_ADL_ACTFOR: PROC (FORM_PTR, HEADER_PTR, MESS_PTR);
20
30
40 DCL  FORM_PTR PTR;                                /* INFOR
      MATION TO MAKE FORMS */
50 DCL  MESS_LINE CHAR (77) CONSTANT INIT ((77)"-");
60 DCL  HEADER_PTR PTR;                              /* POINT
      ER TO THE HEADER OF SCREEN */
70 DCL  MESS_PTR PTR;
80 /*
90      *****
100     *****
110     *****
120     */
113
140          $EST;
-----

```

PR H_ADL_ACTFOR WINDOWING;

L'unité H_ADL_ACTFOR de la bibliothèque LIB est visualisée avec les fonctions WINDOWING. Voir la grille ci-dessous.

```

FSE                                -->:___                                PRINT

      LIBRARY = LINT.CMB.SLLIB
      MEMBER = H_ADL_ACTFOR
-----1-----2-----3-----4-----5-----6-----
10 H_ADL_ACTFOR: PROC (FORM_PTR, HEADER_PTR, MESS_PTR);
20
30
40 DCL  FORM_PTR PTR;                                /* INFOR
50 DCL  MESS_LINE CHAR (77) CONSTANT INIT ((77)"-");
60 DCL  HEADER_PTR PTR;                              /* POINT
70 DCL  MESS_PTR PTR;
80 /*
90      *****
100     *****
110     *****
120     */
130
140          $EST;
150          $ERECTAB;
160          $EFLDTAB 'BASED (CFT_PTR)';
-----1-----2-----3-----4-----5-----6-----

```

2.10 COMMANDE QUIT

La commande QUIT (Q ou /) permet de sortir de FSE.

Grille de guidage

La commande QUIT n'a pas de grille de guidage.

Format en mode ligne

QUIT;

Paramètres

La commande QUIT n'a pas de paramètres.

Exécution de la commande

La commande est exécutée dès sa sélection ou son introduction. Si FSE a été appelé au niveau système, il y a retour au niveau système. Si FSE a été appelé par un autre processeur, il y a retour au niveau commande de ce processeur.

Exemple :

QUIT;

La session FSE est terminée.

2.11 COMMANDE RENAME

La commande RENAME (abréviation RN) permet de changer le nom d'une unité dans la bibliothèque de sortie.

Grille de guidage

```

1/1                                RENAME                                -->:  __
                                rename a member in LIB

MEMBER                            + member to be processed
                                -----
NEW                                + new name of the member
                                -----
REPLACE                            do you want to overwrite?          0
                                -----
-----

```

Format en mode ligne

```

RENAME

    MEMBER = nom-unité

    NEW = nom-unité

    [REPLACE] ;

```

Paramètres

Le paramètre MEMBER spécifie l'unité dont le nom doit être changé.

Le paramètre NEW indique le nouveau nom de l'unité.

Le paramètre REPLACE permet de remplacer dans la bibliothèque de sortie une unité existante par une unité ayant le même nom.

Exemples :

```
RN jh__old NEW=jh_new;
```

Le nom de l'unité jh_old dans la bibliothèque LIB est changée en jh_new.

```
RN jh_old NEW=jh_new REPLACE;
```

Le nom de l'unité jh_old dans la bibliothèque LIB est changée en jh_new. Toute unité portant le même nom (jh_new) dans LIB est remplacée par la nouvelle unité.

2.12 COMMANDE RENUMBER

La commande RENUMBER (abréviation RB) permet de changer les numéros de ligne d'une ou plusieurs unités dans la bibliothèque de sortie.

Grille de guidage

```

1/1                                RENUMBER                                -->:  __
                                renumber member(s) in output lib

MEMBERS                            + candidates for processing
-----
INIT                                starting from                                10
-----
STEP                                increment                                10
-----
BRIEF                              brief report ?                                0
-----
-----

```

Format en mode ligne

RENUMBER

```
[MEMBERS = {nom-unité
            {nom-astérisque}
            {*
            -
            }]
```

```
[INIT = numéro-ligne]
```

```
[STEP = pas-de progression]
```

```
[BRIEF] ;
```

Paramètres

Le paramètre MEMBERS spécifie la ou les unités dont les lignes doivent être renumérotées. Il est possible d'introduire le nom d'une seule unité ou bien d'utiliser la convention astérisque pour désigner un ensemble d'unités. A l'exécution de la commande, il y a visualisation d'une ou plusieurs grilles contenant, pour chaque unité spécifiée, des informations et une zone OMIT. Si la renumérotation des lignes d'une ou plusieurs unités n'est pas souhaitée, remplir les zones OMIT correspondantes. N'importe quel caractère peut être introduit dans la zone OMIT, à l'exception du point d'interrogation (?), qui appelle un texte HELP. Les unités dont la zone OMIT est laissée en blanc sont renumérotées.

Le paramètre INIT indique le numéro à affecter à la première ligne. Si ce paramètre est omis, la valeur implicite est 10.

Le paramètre STEP spécifie le pas de progression à utiliser pour la numérotation des lignes suivantes. Si ce paramètre est omis, le pas de progression utilisé sera 10.

Le paramètre BRIEF demande que seul le nom des unités soit visualisé; les informations telles que la date et l'heure de la dernière modification, le langage et le nombre de lignes ne le seront pas.

Exécution de la commande

Une grille spécifique de la commande RENUMBER est visualisée lorsque les paramètres de la commande ont été acceptés.

Cette grille donne la liste des unités de la bibliothèque de sortie susceptibles d'être renumérotées. Les en-têtes sont :

OMIT	introduire n'importe quel caractère, à l'exception de ?, si vous ne voulez pas que les lignes de l'unité soient renumérotées.
NAMES	nom de l'unité.
SAVED	date et heure de la dernière modification, sous la forme mm/jj/aa hh:mm
LANG	langage de l'unité.
LINES	nombre de lignes de l'unité.

Lorsque la zone OMIT des unités à ne pas renuméroter a été remplie, appuyer sur la touche TRANS. Une liste indiquant les unités renumérotées et les unités non renumérotées s'affiche. La grille suivante (s'il y en a une) des unités susceptibles d'être renumérotées est ensuite visualisée.

Pour mettre fin à l'exécution de la commande RENUMBER, introduire une barre oblique (/) dans la zone "-->:" ; les lignes des unités dont le nom est visualisé ne seront pas renumérotées.

Description des commandes

Exemple :

```
RB H_*;
```

Toutes les unités de la bibliothèque LIB dont le nom commence par 'H_' seront susceptibles d'être renumérotées. Un exemple d'utilisation de la grille pendant l'exécution de cette commande est donné ci-dessous.

```
FSE                                -->: __                                RENUMBER

LIBRARY = TEMP.SLLIB$TEMPRY

OMIT---NAMES---                    ----SAVED-----    -LANG-    -LINES-

N   H_ADL_ECOMPL                    01/28/84 10:58      GPL        2192
N   H_ADL_EDIR                      03/26/83 14:28      GPL         599
N   H_ADL_ELOAD                     03/26/83 14:29      GPL         270
N   H_ADL_EMAIN                     03/26/83 14:30      GPL         251
N   H_ADL_EPRISF                    03/26/83 14:30      GPL         274
N   H_ADL_ESAVE                     10/18/83 17:05      GPL       2349
N   H_ADL_ETEST                     11/19/83 17:41      GPL         500
    H_ADL_EUNSVE                    03/26/83 14:30      GPL         180
    H_ADL_EXANAL                    10/13/83 18:52      GPL         674
    H_ADL_MGPL                      10/13/83 15:36      ADL         209
    H_ADL_TOKGEN                    06/09/83 12:09      GPL         791
    H_BIN_A_CCBX                    04/02/83 18:45      ADL         134
    H_BIN_A_CCOB                    04/02/83 18:46      ADL         138
    H_BIN_A_CDAT                    12/06/83 18:27      DAT         283
- - - - -
```

Dans cet exemple, la commande RENUMBER H_* a été émise. La grille ci-dessus s'affiche. La lettre N est introduite dans la zone OMIT de chaque unité qui ne doit pas être renumérotée. L'opération est validée par un appui sur la touche de transmission (TRANS). Le résultat s'affiche sur la grille ci-dessous :

```
FSE                                -->: __                                RENUMBER

LIBRARY = TEMP.SLLIB$TEMPRY

OMIT---NAMES---                    ----SAVED-----    -LANG-    -LINES-

N   H_ADL_ECOMPL                    NOT RENUMBERED
N   H_ADL_EDIR                      NOT RENUMBERED
N   H_ADL_ELOAD                     NOT RENUMBERED
N   H_ADL_EMAIN                     NOT RENUMBERED
N   H_ADL_EPRISF                    NOT RENUMBERED
N   H_ADL_ESAVE                     NOT RENUMBERED
N   H_ADL_ETEST                     NOT RENUMBERED
    H_ADL_EUNSVE                    RENUMBERED
    H_ADL_EXANAL                    RENUMBERED
    H_ADL_MGPL                      RENUMBERED
    H_ADL_TOKGEN                    RENUMBERED
    H_BIN_A_CCBX                    RENUMBERED
    H_BIN_A_CCOB                    RENUMBERED
    H_BIN_A_CDAT                    RENUMBERED
- - - - -
```


3. Session MODIFY

Ce chapitre expose les concepts de base de la session MODIFY.

3.1 PRESENTATION

FSE permet la création ou la modification d'unités de bibliothèque origine (SL). Des opérations simples ou complexes peuvent être exécutées sur ces unités. Elles sont spécifiées au moyen d'un jeu de demandes qui constituent un langage de manipulation de texte puissant.

La présentation des concepts et des possibilités de FSE est faite de manière progressive, l'utilisateur pouvant arrêter sa lecture à n'importe quel endroit du chapitre, quand il estime disposer ces informations dont il a besoin.

Une fois la commande MODIFY émise, les demandes de lecture, de modification et de sauvegarde des unités de bibliothèque origine peuvent être introduites.

Trois méthodes sont possibles : l'utilisation de demandes "ligne", l'utilisation de demandes "écran" et les modifications locales sur écran.

Les modifications ne sont pas appliquées directement à l'unité cible mais s'effectuent dans un espace de travail temporaire appelé "tampon". La structure d'un tampon est présentée plus loin dans ce chapitre.

Pour sortir de la session MODIFY, frapper QUIT, Q ou / dans la zone "request" (demande).

3.2 DESCRIPTION DES GRILLES

3.2.1 Grille principale

La première grille qui s'affiche au début d'une session MODIFY est la grille principale (MAIN), qui permet d'introduire n'importe quel type de demande.

L'utilisateur peut travailler en mode novice ou en mode expert. Ces deux modes se différencient par la manière dont les demandes sont introduites au niveau de la grille principale.

La grille principale est divisée en trois zones :

Zone	Lignes
Zone en-tête	1
Zone texte	2 à 20
Zone demande ("request")	22 à 23

Chaque zone est décrite ci-après.

Exemple de grille principale en mode expert :

```

b=MAIN m=LIB: DEM DAT > ln=2470
TOP
ctl _____
=====
1 =====
2 =====
3 =====
4 =====
4.1 =====
4.2 =====
4.3 =====
5 =====
6 =====
7 =====
8 =====
9 =====
10 =====
11 =====
12 =====
13 =====
14 =====
15 =====
16 =====

requests_____
_____
    
```

Introduire les demandes, avec leurs paramètres, dans la zone "requests". Il est possible de spécifier plusieurs demandes, en les séparant par un point-virgule ";".

Session Modify

Exemple de grille principale en mode novice :

```

b=MAIN m=LIB: DEM DAT > ln=2470

                                TOP                                ctl _____

1  ===== text =====
2  ===== text =====
3  ===== text =====
4  ===== text =====
4.1 ===== text =====
4.2 ===== text =====
4.3 ===== text =====
5  ===== text =====
6  ===== text =====
7  ===== text =====
8  ===== text =====
9  ===== text =====
10 ===== text =====
11 ===== text =====
12 ===== text =====
13 ===== text =====
14 ===== text =====
15 ===== text =====
16 ===== text =====

requests _____ addr _____
param 1 _____ param 2 _____ param 3 _____

```

Introduire la demande dans la zone "request", le paramètre adresse, s'il y en a un, dans la zone "addr" et les paramètres supplémentaires éventuels dans les zones "param1", "param2" et "param3".

3.2.1.1 Zone en-tête

b=<tampon> m=<unité> <lang> <bornes> <cntxt> <dir> ln=<nbrel> ctl=<zgest>

Signification :

b=<tampon> :	nom du tampon.
m=<unité> :	"no member" si aucune unité n'a été associée, l'identification de la bibliothèque et l'unité associée dans les autres cas. Exemple : m=LIB:DOC désigne l'unité DOC qui se trouve dans la bibliothèque LIB. (Pour plus de détails, voir le paragraphe 3.11.2).
<lang> :	langage.
<bornes> :	indique les bornes définies par la dernière demande BNDS. (Cette information n'est pas affichée si aucune demande BNDS n'a été émise).
<cntxt> :	expression ou identificateur de contexte (n'apparaît qu'en mode mise à jour de contexte).
<dir> :	{ > < } direction implicite pour le saut de page.
ln=<nbrel> :	nombre de lignes du tampon.
ctl=<zgest> :	zone de gestion.

3.2.1.2 Zone texte

Cette zone permet d'afficher 20 lignes du tampon courant. Au moment de la visualisation du contenu du tampon, les lignes virtuelles TOP et BOTTOM se placent respectivement avant la première ligne et après la dernière ligne du tampon.

Le groupe de 20 lignes peut être considéré comme une "fenêtre" permettant d'afficher le contenu du tampon et qui se déplace vers le haut ou vers le bas selon les demandes faites.

Chaque ligne du texte est précédée d'un numéro. Si le numéro est trop long pour être visualisé, un astérisque s'affiche.

3.2.1.3 Zone "request"

Cette zone est utilisée pour introduire les demandes "ligne".

- (1) En mode expert, la zone "requests" fait 148 caractères et le nombre de demandes qui peuvent être introduites à la fois n'est limité que par leur longueur respective.
- (2) En mode novice, il y a cinq subdivisions : "request" (demande), "addr" (adresse), "param1", "param2" et "param3". Une seule demande à la fois peut être introduite.

3.2.1.4 Opérations

Les opérations qui peuvent être demandées à partir de la grille principale sont les suivantes :

Au niveau de la zone de gestion :

- ? appel de texte HELP
- / effacement de toutes les zones, y compris la zone "request",
rejet des éventuelles modifications locales et arrêt de toute
opération sur les tampons.
- blanc : reprise (ou lancement) de l'exécution normale

Toute autre entrée est signalée comme étant une erreur sur le compte rendu d'exécution.

Au niveau des autres zones :

- Si la zone contient la valeur voulue, la laisser telle quelle.
- Si la zone est vide ou contient une valeur qui n'est pas la bonne, frapper la valeur voulue, qui écrase éventuellement la valeur incorrecte.
- Pour obtenir une aide (HELP), introduire un point d'interrogation (?).

Les valeurs erronées ou non valables sont affichées en vidéo inverse et un message de diagnostic est émis.

3.2.2 Grille de guidage

Le deuxième type de grille qui peut s'afficher est la grille de guidage. Cette grille est propre à une demande et permet d'introduire ou de modifier les paramètres de cette demande.

Cette grille ne présente que quelques différences mineures avec la grille principale du mode novice :

- Seuls les paramètres concernant la demande sont visualisés, les autres sont verrouillés.
- Les noms des paramètres sont propres à la demande et non pas généraux.
- La zone "request" est remplie puis verrouillée.
- Les demandes "écran" et les modifications locales ne peuvent pas être introduites en mode écran, et l'accès à la zone texte est verrouillé.

Exemple de grille de guidage :

```

b=MAIN m=LIB: DEM DAT   > ln=2470
                                TOP                               ctl _____
=====
1  =====                    text =====
2  =====                    text =====
3  =====                    text =====
4  =====                    text =====
4.1 =====                  text =====
4.2 =====                  text =====
4.3 =====                  text =====
5  =====                    text =====
6  =====                    text =====
7  =====                    text =====
8  =====                    text =====
9  =====                    text =====
10 =====                   text =====
11 =====                   text =====
12 =====                   text =====
13 =====                   text =====
14 =====                   text =====
15 =====                   text =====
16 =====                   text =====

requests S _____  addr _____
old_strg _____    new__ strg _____
    
```

3.2.2.1 Opérations

Zone de gestion (ctl)

Les opérations qu'il est possible de demander dans la zone de gestion (ctl) de la grille de guidage sont les suivantes :

- ? demande de texte HELP
- / effacement de toutes les zones, y compris la zone "request", et arrêt de l'exécution des demandes suivantes (que celles-ci soient dans les tampons ou qu'elles aient été introduites auparavant dans la zone "request").
- ; sortie de la grille de guidage sans pour cela perdre les demandes courante et suivantes. La grille principale s'affiche alors et contient dans la zone "request" les demandes "ligne" non exécutées. Lors de l'exécution d'une demande qui se trouve dans le tampon, la zone "request" est une copie de la ligne du tampon en cours d'exécution

- blanc : demande de la reprise de l'exécution.

Toute autre entrée est signalée dans le compte rendu d'exécution comme étant une erreur.

Autres zones

- si la zone contient déjà la valeur voulue, la laisser telle quelle.
- si la zone est vide ou contient une valeur qui n'est pas la bonne, frapper la valeur voulue qui écrasera éventuellement la valeur antérieure.
- pour obtenir une aide (HELP), introduire un point d'interrogation (?).

Les valeurs erronées ou non valables sont affichées en vidéo inverse et un message de diagnostic est émis.

Tout comme avec la grille principale, des paramètres plus longs que la normale peuvent être introduits ou visualisés en utilisant des grilles "grande zone". Seuls les grilles correspondant à un paramètre valable pour l'opération sont disponibles.

Citons, à titre d'exemple, l'utilisation d'une chaîne de caractères relativement longue pour la demande SP (coupure de ligne) : trois grilles sont disponibles, une pour le nom de la demande, une pour l'adresse et une pour la chaîne de caractères (paramètre).

Ces grilles "grande zone" sont affichées dès qu'une valeur de paramètre est trop longue pour être visualisée dans une zone de taille normale, ou si "+" est détecté à la place d'un paramètre. L'utilisateur peut obtenir ce type de grille en introduisant le signe + à la place de n'importe quel paramètre.

3.3 VISUALISATION DE TEXTE

Quelle que soit la grille, la visualisation d'un texte dans la zone texte se fait de telle manière que la ligne courante apparaît approximativement au milieu de l'écran, sauf si la nouvelle ligne courante apparaissait déjà sur la grille précédente. Dans ce dernier cas, le positionnement du texte dans la zone texte se fait de façon à minimiser les déplacements du texte.

3.4 SAUT DE PAGE

Lorsque le texte visualisé a été modifié, la grille courante s'affiche de nouveau, ou bien un saut de page automatique est exécuté dans la direction spécifiée en dernier. La direction (> ou <) est indiquée au niveau de l'en-tête.

3.5 INTRODUCTION DES DEMANDES

Les demandes peuvent être introduites et exécutées en mode ligne ou en mode écran :

- (1) En mode ligne, les demandes sont introduites uniquement dans la zone "request". (Les lignes à modifier ou à balayer sont spécifiées en utilisant les techniques d'adressage décrites au paragraphe 3.7).
- (2) En mode écran, les modifications sont apportées de trois façons :
 - en modifiant localement chaque ligne visualisée à l'écran.
 - en introduisant dans la zone "numéro de ligne" (appelée également "marge") de n'importe quelle ligne visualisée, la demande "écran" à exécuter pour cette ligne.
 - en utilisant, comme en mode ligne, des demandes "ligne".

3.5.1 Demandes "ligne"

Chaque demande "ligne" est spécialisée et correspond à une seule fonction, mais toutes les demandes respectent les mêmes règles :

- (1) Une demande "ligne" se compose de deux parties :
 - le nom de la demande qui identifie la fonction.
 - les paramètres qui fournissent des informations complémentaires sur l'opération demandée.
- (2) Une demande peut comporter quatre paramètres au maximum :
 - un paramètre adresse, au cas où les opérations sur les lignes doivent être faites de manière sélective.
 - et jusqu'à trois paramètres supplémentaires.

Les demandes "ligne" peuvent être introduites de trois façons différentes :

- En utilisant la grille principale du mode novice : chaque élément de la demande (le nom de la demande et les paramètres) doit être introduit dans la zone correspondante. Les zones qui sont sans objet pour la demande doivent être laissées en blanc.
- En utilisant la grille de guidage : les zones spécifiques de la demande sont visualisées et doivent obligatoirement être remplies (sauf si les paramètres correspondants sont facultatifs).
- En mode libre, en utilisant la grille principale du mode expert, ou en exécutant la demande à partir d'un tampon.

Remarque : Le nom de la demande peut être spécifié sous la forme "nom?" ou "?nom" dans tous les cas où un nom de demande est accepté. Si

"nom?" est introduit, le guidage correspondant s'affiche avec les valeurs des paramètres, si elles ont été fournies. Si c'est "?nom" qui est introduit, le texte HELP s'affiche en premier, avant le guidage.

Exemples :

```
A? . ;
?S
```

Quel que soit le mode, il est possible d'introduire un point d'interrogation (?) ou un signe "plus" (+) à la place de n'importe quel paramètre afin qu'il soit visualisé avec le texte HELP correspondant (voir grille "grande zone").

Exemples :

```
S . + +
S . ? 'profil'
```

La séquence d'échappement [V permet de fournir comme paramètre d'une demande la valeur d'une variable GCL.

Le format de la séquence d'échappement [V est :

```
[V (#nom-variable-système)
ou
[V ([%]nom-variable-globale)
```

La séquence d'échappement de variable GCL remplace un paramètre entier (et non pas une partie seulement du paramètre).

Exemple :

```
IL3 [V(#SINLB(2)) ;
```

3.5.2 Mode libre

L'utilisateur peut introduire plusieurs demandes sur la ligne (ligne "request" ou ligne du tampon d'exécution).

Dans ce cas, chaque demande est séparée de la demande précédente par un point-virgule.

Exemple :

```
demande-1; demande-2; demande-3
```

Les paramètres d'une demande sont positionnels, c'est-à-dire qu'ils doivent apparaître dans l'ordre indiqué dans ce manuel.

Session Modify

Le nom de la demande est séparé du premier paramètre par au moins un espace . Les paramètres doivent être séparés les uns des autres par des espaces ou des virgules.

Les virgules peuvent être précédées ou suivies d'espaces. Plusieurs espaces consécutifs sont l'équivalent d'un seul espace .

Si un paramètre est inutile (autrement dit, si la valeur implicite convient), introduire deux virgules à la suite l'une de l'autre.

Exemples :

```
RI . unité , , 1:50;
```

Si les paramètres qui ne doivent pas être modifiés sont en fin de ligne, les virgules peuvent être omises.

```
RI . unité;
```

Si différentes demandes ont été introduites, elles sont exécutées une par une. Si une erreur est détectée, l'exécution de la demande est immédiatement interrompue et un message d'erreur s'affiche. Les demandes qui n'ont pas encore été exécutées sont elles aussi visualisées à l'écran.

Si des paramètres sont incorrects ou si une aide est nécessaire pour une demande particulière, une grille de guidage s'affiche et permet d'introduire ou de modifier les paramètres de la demande.

3.6 NUMEROTATION DES LIGNES

L'unité à lire à partir d'une bibliothèque SL doit être en format SSF (System Standard Format). Dans ce format, chaque ligne a un en-tête SSF qui contient les informations suivantes :

- (1) Un numéro de ligne, sous forme d'un entier.
- (2) Un drapeau "ligne_incluse".
- (3) Un drapeau "ligne_développée par le macro-processeur".

Lors de la visualisation du texte au cours de la session MODIFY, FSE utilise les deux formats suivants selon le contenu de l'en-tête SSF.

Si les deux drapeaux ont la valeur "faux" le numéro de ligne est imprimé sous la forme d'un entier.

Si l'un des drapeaux a la valeur "vrai", le numéro de ligne est imprimé sous la forme n.m, n étant le dernier numéro de la ligne pour laquelle les deux drapeaux étaient "faux", et m étant le numéro d'ordre de la ligne courante par rapport à la ligne n.

Exemple :

```
325    ligne normale 1
325.1  première ligne avec un drapeau "vrai"
325.2  deuxième ligne avec un drapeau "vrai"
326    ligne normale 2
327    ligne normale 3
```

Chaque fois qu'une demande d'effacement, d'ajout ou d'insertion est exécutée, les numéros de la séquence modifiée sont automatiquement recalculés.

Exemple :

Après remplacement de la ligne numérotée 325.1 par deux nouvelles lignes, la séquence modifiée est :

```
325    ligne normale 1
325.1  première nouvelle ligne
325.2  deuxième nouvelle ligne
325.3  deuxième ligne avec un drapeau "vrai"
326    ligne normale 2
327    ligne normale 3
```

Session Modify

Après transfert des lignes 325.3 et 326 après la ligne 325 la séquence est :

```
325   ligne normale 1
325.1 deuxième ligne avec un drapeau "vrai"
325.2 ligne normale 2
325.3 première nouvelle ligne
325.4 deuxième nouvelle ligne
327   ligne normale 3
```

Exemple :

Pour la sauvegarde du contenu d'un tampon dans une unité, il faut respecter les règles suivantes :

- (1) Si le numéro de ligne est un entier, celui-ci est placé dans la zone en-tête SSF et les deux drapeaux sont forcés à "faux".
- (2) Si le numéro de ligne a le drapeau "ligne-incluse" ou si la ligne a été insérée lors d'une session MODIFY et renumérotée, le numéro est placé dans la zone numéro de ligne de l'en-tête SSF et le drapeau "ligne incluse" est forcé à "vrai".

Si le numéro de ligne a été détecté et si le drapeau correspondant est "ligne-développée" mais qu'il n'y a pas eu de renumérotation de cette ligne, l'en-tête SSF reste tel qu'il était au moment de la lecture.

Remarque : Une ligne avec un numéro de ligne égal à zéro et deux drapeaux ayant la valeur "faux" est considérée comme une ligne ayant le drapeau "ligne-développée".

3.7 ADRESSAGE DE LIGNE

L'adressage de ligne est la fonction qui permet la sélection des lignes sur lesquelles les demandes doivent porter.

La plupart des demandes peuvent s'appliquer à n'importe quel ensemble de lignes (contiguës ou non).

D'autres demandes (par exemple, la demande de sauvegarde SV) ne peuvent traiter qu'un ensemble de lignes contiguës.

Quelques demandes ne peuvent porter que sur une seule ligne (par exemple, la demande L). Enfin, certaines demandes ne s'appliquent pas à des lignes et ne nécessitent donc aucune adresse (comme DISP).

Le type d'adressage est indiqué séparément pour chaque demande.

Il existe trois techniques de base pour l'adressage d'UNE ligne dans un tampon :

- l'adressage par numéro de ligne,
- l'adressage par application d'un incrément,
- l'adressage par identification d'un profil .

Les paragraphes suivants décrivent ces techniques et montrent comment elles peuvent se combiner.

3.7.1 Adressage par numéro de ligne.

Chaque ligne d'un tampon peut être identifiée par son numéro de ligne.

Exemple :

10	indique la ligne portant le numéro de ligne 10
1290.1	indique la ligne portant le numéro virtuel 1290.1 (le numéro SSF étant 1 et le numéro de la ligne précédente 1290)

Il est possible que plusieurs lignes aient le même numéro de ligne (dans le cas, par exemple, où la totalité du tampon n'a pas été renumérotée). La ligne concernée est alors la première ligne ayant le numéro de ligne donné qui est détectée.

La recherche du numéro de ligne s'effectue de la ligne courante à la dernière ligne du tampon, puis de la première ligne du tampon jusqu'à la ligne courante, si la ligne recherchée n'a pas été trouvée avant.

A noter que la recherche est optimisée lorsque les numéros se présentent dans le tampon en ordre croissant.

Session Modify

Au niveau de FSE, la ligne courante, la première et la dernière ligne d'un tampon sont représentées par les caractères suivants :

.	(point)	ligne courante
^	(négation)	première ligne
\$	(dollar)	dernière ligne

Ces trois caractères peuvent être utilisés à la place du numéro de ligne.

3.7.2 Adressage par application d'un incrément

Une ligne peut être désignée par un incrément, qui se présente sous la forme d'un entier avec signe.

Exemple :

-1	ligne située avant la ligne courante
+3	troisième ligne après la ligne courante

3.7.3 Adressage par identification d'un profil

La nième occurrence d'une chaîne de caractères est recherchée en avant ou en arrière dans les lignes du tampon. En recherche avant, la recherche s'effectue de la ligne courante à la dernière ligne du tampon. En recherche arrière, la recherche s'effectue de la ligne courante à la première ligne du tampon. La recherche s'arrête à la nième ligne qui contient la chaîne de caractères.

Le sens de la recherche est spécifié par les signes - ou + (+ est le sens implicite). Le numéro d'occurrence est indiqué sous la forme [#n, n étant le numéro souhaité (le numéro 1 est le numéro implicite). On trouve ensuite l'expression régulière. Cet ordre est obligatoire.

Exemple :

/xxx/	équivalent à +[#1/xxx/ et signifie : recherche en avant de la première ligne contenant la chaîne de caractères 'xxx'.
-/yyy/	équivalent à -[#1/yyy/ et signifie : recherche en arrière de la première ligne contenant la chaîne de caractères 'yyy'.

3.7.3.1 Expression régulière

Une expression régulière est composée d'une ou deux chaînes régulières reliées entre elles par un opérateur logique au maximum.

Citons, à titre d'exemple, l'opérateur unaire [^ (non) et les opérateurs binaires [^ (mais pas), [& (et), [! (ou).

Exemple :

-[#3/reg1/[&/reg2/ signifie recherche en arrière (c'est-à-dire de la ligne courante à la première ligne) de la 3ème ligne correspondant à reg1 et reg2.

3.7.3.2 Chaîne régulière

Une chaîne régulière peut contenir n'importe quels caractères. Elle est délimitée par les caractères barre oblique (/) et apostrophe (').

Certains caractères ont une signification particulière :

/ (barre oblique)	Deuxième délimiteur de la chaîne si le premier est une barre oblique.
' (apostrophe)	Deuxième délimiteur de la chaîne si le premier est une apostrophe.
* (astérisque)	N'importe quelle occurrence (y compris zéro occurrence) du caractère précédent éventuel.
^ (négation)	La chaîne spécifiée est au début de la ligne. Lorsqu'il est le premier caractère de l'expression régulière, ^ désigne le caractère virtuel précédant le premier caractère de la ligne.
\$ (dollar)	La chaîne spécifiée se situe à la fin de la ligne. Lorsqu'il est le dernier caractère de l'expression régulière, \$ désigne le caractère virtuel suivant le dernier caractère de la ligne.
. (point)	N'importe quel caractère d'une ligne.

Exemples :

/A /	la ligne recherchée contient la lettre A suivie d'un espace.
/^ABC/	la ligne recherchée commence par la chaîne ABC.
/ABC\$/	la ligne recherchée se termine par la chaîne ABC.

Session Modify

<code>/AB*C/</code>	la ligne recherchée contient l'une des chaînes AC, ABC, ABBC ou ABBBC, etc.
<code>/IN..TO/</code>	la ligne recherchée contient la chaîne IN, suivie de deux caractères quelconques, eux-mêmes suivis de la chaîne TO.
<code>/^ABC.*DEF\$/</code>	la ligne recherchée commence par ABC et se termine par DEF.

Certaines séquences ont une signification particulière :

<code>[X</code> ou <code>[x</code>	sert à spécifier un caractère qui n'a pas de représentation graphique. La séquence <code>[Xhh</code> (hh étant deux caractères hexadécimaux) représente le caractère en question.
<code>[C</code> ou <code>[c</code>	cette séquence annule la signification particulière des caractères suivants : <code>^ . * X [C / ' </code>

Exemples :

<code>/[C^/</code>	la ligne recherchée contient le caractère " ^ "
<code>/[C/[C*/</code>	la ligne recherchée contient la chaîne " /* "
<code>/[C[Xaa/</code> ou <code>/[CXaa/</code>	la ligne recherchée contient " [Xaa "
<code>/[C[C/</code> ou <code>/[[cC/</code>	la ligne recherchée contient " [C "

Remarque : L'adressage sur la base d'un profil donné s'effectuant par la sélection d'une chaîne de caractères, il y a plusieurs possibilités dans le cas où une expression régulière est utilisée comme premier paramètre d'une demande de remplacement ou de coupure de ligne. A ce moment là, la convention suivante est adoptée : la chaîne de caractères sélectionnée est celle qui est à l'extrême gauche.

Exemples :

<code>/b*c/</code>	dans la chaîne de caractères 'aabbcd', la chaîne sélectionnée est 'bbbc'.
<code>/B* ..cd/</code>	dans la chaîne de caractères 'aabbcd' la chaîne sélectionnée est 'bbbc'.

- 2) Lorsque la chaîne régulière se termine par un astérisque (*), la sélection de la chaîne de caractères s'arrête juste avant le premier caractère qui ne correspond pas.

Exemples :

/ab*/ dans la chaîne de caractères 'abbcd', la chaîne de caractères sélectionnée est 'abb'.
/^ */ la chaîne de caractères sélectionnée contient tous les espaces trouvés en début de ligne.

A noter que l'expression régulière /a*/ est égale à l'expression régulière /^a*/; de même l'expression régulière / */ équivaut à l'expression régulière /^ */ , seuls les espaces en début de ligne étant sélectionnés.

3.7.4 Combinaison des méthodes d'adressage

Les trois techniques de base décrites ci-dessus peuvent être combinées. Le numéro de ligne ne doit apparaître qu'une fois en début de ligne.

Chacun des éléments de l'adresse composée est successivement évalué de gauche à droite.

Exemples :

60+5 la ligne recherchée se trouve cinq lignes après la ligne ayant le numéro 60.
.+4/PPP/ la ligne recherchée contient PPP; la recherche de cette ligne commence quatre lignes après la ligne courante.
/IOF/+5 la ligne recherchée est la cinquième ligne après la ligne contenant IOF.
/TDS//IOF/-1 la ligne recherchée précède la ligne contenant IOF qui vient après une ligne contenant TDS.

3.7.4.1 Adressage d'une plage de lignes

Une plage de lignes peut être constituée de deux lignes spécifiées séparées par le caractère "deux-points" (:)

Les deux lignes spécifiées sont la première et la dernière ligne de la plage. Le calcul de la seconde ligne est totalement indépendant de la première. La valeur de la ligne courante reste inchangée jusqu'à l'exécution de la demande.

Exemple :

-1 : 'ABC'

La première ligne est la ligne placée avant la ligne courante et la dernière ligne est la première ligne contenant 'ABC' détectée à partir de la ligne courante.

Toute les demandes effectuées sur une plage de lignes peuvent l'être aussi sur une seule ligne.

3.7.4.2 Adressage d'un ensemble de lignes

L'adressage d'un ensemble de lignes se définit par deux techniques supplémentaires :

- 1) Ensemble de lignes délimitée par deux lignes et comportant le profil correspondant à une expression régulière donnée.

Exemples :

1:10='profil'	l'ensemble de lignes spécifié est constitué des lignes qui, entre la ligne 1 et la ligne 10, contiennent le profil indiqué.
1:5='profil_1 ' [!'profil_2'	l'ensemble de lignes spécifié est constitué des lignes qui, entre la ligne 1 et la ligne 5, contiennent soit "profil_1" soit "profil_2".

Les règles de spécification et d'évaluation de la première et de la dernière ligne sont les mêmes que pour une plage de lignes. Un caractère "=" sert à introduire une expression régulière.

- 2) Ensemble de lignes spécifié en mettant sous forme de liste n'importe quelle combinaison précédemment décrite.

Exemple :

(1 4:5, 30:35)	les lignes spécifiées sont les lignes 1,4,5 et les lignes 30 à 35.
----------------	--

Les éléments d'une liste doivent être séparés par une virgule ou par un ou plusieurs espaces. Des espaces consécutifs équivalent à un seul espace et une virgule peut être elle-même précédée et suivie d'espaces.

La liste doit obligatoirement être entre parenthèses et peut contenir n'importe quel chiffre.

3.7.5 Règles de spécification des adresses

<ligne_unique>	:= <ligne>
<plage>	:= <ligne>:<ligne>
<ensemble>	:= <ligne>:<ligne>=<expr_rég> ! ({ <ligne> ! <ligne>:<ligne> ! <ligne>:<ligne>=<expr_rég> } ...)
<expr_rég>	:= <expr_simple> ! <et_expr> ! <ou_expr> ! <mais_pas_expr> ! <pas_expr>
<expr_simple>	:= <chaîne régulière>
<et_expr>	:= <chaîne_rég.> [& <chaîne_rég.>
<ou_expr>	:= <chaîne_rég.> [! <chaîne_rég.>
<mais_pas_expr>	:= <chaîne_rég.> [^ <chaîne_rég.>
<pas_expr>	:= [^ <chaîne_régulière>
<chaîne_régulière>	: voir le paragraphe précédent.
<ligne>	:= { <numéro_ligne> ! ^ ! \$! . } [[<incrément>] [<nième_occurrence_expr_rég>]]...
<numéro_ligne>	:= <entier> [.<entier>]
<incrément>	:= {+ ! -} <entier>
<nième_occurrence_expr_rég>	:= {+ ! -} [#<nième_occ>] <expr_rég>

3.8 MODE SAISIE

FSE affiche huit zones en entrée :

4 zones de saisie de 70 caractères.

4 zones "marge" de 7 caractères.

Les données sont introduites dans les zones de saisie.

Les caractères non graphiques sont acceptés sous la forme [Xhh, hh étant le code hexadécimal du caractère.

Les espaces de droite sont supprimés.

Les lignes d'entrée dont la longueur est supérieure à 255 caractères sont tronquées.

Dans les zones "marge", l'utilisateur doit préciser si les données des zones de saisie constituent le début d'un nouvel article (ce qui est la valeur implicite) ou la continuation de l'article précédent (caractère & dans la zone marge).

Pour sortir du mode saisie :

- Introduire un point-virgule (;) dans la zone de gestion (ctl), la zone d'entrée étant alors traitée avant de sortir du mode saisie.
- introduire / dans la zone de gestion (ctl), la zone d'entrée n'étant pas traitée et la session saisie prenant fin.

Session Modify

Exemple de session avec demande A (Adjonction) :

```

b=MAIN m=LIB: DEM DAT > ln=2470
                                                    ctl _____
    100 THIS IS RECORD NO          100
    110 THIS IS RECORD NO          110
    120 THIS IS RECORD NO          120
    130 THIS IS RECORD NO          130
    140 THIS IS RECORD NO          140
    150 THIS IS RECORD NO          150
    160 THIS IS RECORD NO          160
    170 THIS IS RECORD NO          170
    180 THIS IS RECORD NO          180
    190 THIS IS RECORD NO          190
    200 THIS IS RECORD NO          200
>>>> 210 THIS IS RECORD NO          210
    220 THIS IS RECORD NO          220
    230 THIS IS RECORD NO          230
    240 THIS IS RECORD NO          240
    250 THIS IS RECORD NO          250
    260 THIS IS RECORD NO          260
    270 THIS IS RECORD NO          270
    280 THIS IS RECORD NO          280
    290 THIS IS RECORD NO          290
        THIS IS RECORD NO
        THIS IS RECORD NO

requests A 200 _____
_____

```

```

b=MAIN m=LIB: DEM DAT > ln=2470
                                                    ctl _____
    130 THIS IS RECORD NO          130
    140 THIS IS RECORD NO          140
    150 THIS IS RECORD NO          150
    160 THIS IS RECORD NO          160
    170 THIS IS RECORD NO          170
    180 THIS IS RECORD NO          180
    190 THIS IS RECORD NO          190
    200 THIS IS RECORD NO          200
        This is the first record appended _____
        This is the second record which is more than 70 _____
        characters long
    & 210 This is the third record appended _____
    220 THIS IS RECORD NO          220
    230 THIS IS RECORD NO          230
    240 THIS IS RECORD NO          240
    250 THIS IS RECORD NO          250
    260 THIS IS RECORD NO          260
    270 THIS IS RECORD NO          270
    280 THIS IS RECORD NO          280
        THIS IS RECORD NO
        THIS IS RECORD NO

requests _____
_____

```

FSE - Editeur Plein Ecran

Une fois ces trois articles introduits, il faut appuyer sur la touche de transmission (TRANS) afin d'obtenir la grille de saisie suivante et introduire les autres articles.

```
b=MAIN m=LIB: DEM DAT > ln=2470
                                     ctl _____
170 THIS IS RECORD NO                170
180 THIS IS RECORD NO                180
190 THIS IS RECORD NO                190
200 THIS IS RECORD NO                200
200.1 This is the first record appended
200.2 This is the second record which is more than 70
      characters long
200.3 This is the third record appended
      This is the fourth record appended
      This is the fifth and last record appended_____
                                     _____
210 _____                        210
220 THIS IS RECORD NO                220
230 THIS IS RECORD NO                230
240 THIS IS RECORD NO                240
250 THIS IS RECORD NO                250
260 THIS IS RECORD NO                260
270 THIS IS RECORD NO                270
280 THIS IS RECORD NO                280
      THIS IS RECORD NO
requests _____
      _____
```

Les deux derniers articles ont été introduits et un appui sur la touche HOME (ORIGINE) a placé un point-virgule (;) dans la zone de gestion (ctl), ce qui permet de sortir du mode saisie.

Session Modify

La grille suivante s'affiche ensuite :

```
b=MAIN m=LIB: DEM DAT > ln=2470

ctf _____

170 THIS IS RECORD NO 170
180 THIS IS RECORD NO 180
190 THIS IS RECORD NO 190
200 THIS IS RECORD NO 200
200.1 This is the first record appended
200.2 This is the second record which is more than 70
characters long
200.3 This is the third record appended
200.4 This is the fourth record appended
>> 200.5 This is the fifth and last record appended
210
220 THIS IS RECORD NO 210
230 THIS IS RECORD NO 220
240 THIS IS RECORD NO 230
250 THIS IS RECORD NO 240
260 THIS IS RECORD NO 250
270 THIS IS RECORD NO 260
280 THIS IS RECORD NO 270
290 THIS IS RECORD NO 280
300 THIS IS RECORD NO 290
THIS IS RECORD NO 300

requests _____
```

- Remarques :**
- 1) Les lignes ajoutées sont automatiquement renumérotées de 200.1 à 200.5.
 - 2) La nouvelle ligne courante est la dernière ligne ajoutée.

3.9 MODE FORMATE

Certaines demandes utilisent une application fournie par le système et correspondant au langage utilisé, pour introduire, mettre à jour ou imprimer des articles en mode formaté.

Ces demandes sont les suivantes :

- AF, IF et CF (Adjonction, insertion et modification en mode formaté).
- ULF (Mise à jour de ligne en mode formaté).
- PF (Impression en mode formaté).

Les formats de langage standard listés ci-dessous sont acceptés.

Langage	Abrév.	Format (les numéros représentent les colonnes)
COBOL	COB	1-6 : n° de ligne, 7-80 : texte
COBOLX	CBX	1-6 : n° de ligne, 7-72 : texte
DATASSF	DAT	pas de n° de ligne, articles ayant la longueur réelle du texte (maxi : 255 car.)
FORTRAN	FOR	pas de n° de ligne, 1-72 : texte
GPL	GPL	1-6 : n° de ligne, 7-80 : texte
JCL (*)	JCL	pas de n° de ligne, articles ayant la longueur réelle du texte (maxi : 255 car.)
CMD (*)	CMD	pas de n° de ligne, articles ayant la longueur réelle du texte (maxi : 255 car.)
PASCAL	PAS	pas de n° de ligne, articles ayant la longueur réelle du texte (maxi : 255 car.)
C_LANG	CL	pas de n° de ligne, articles ayant la longueur réelle du texte (maxi : 255 car.)

(*) Le format de ces langages est en fait le format de données libre. Il est donc préférable d'utiliser des demandes non formatées pour les unités de ce type.

3.10 CHANGEMENT DES CARACTERES DE COMMANDE DE FSE ('^', '\$' ET '[').

Les trois caractères '^', '\$' et '[' de FSE peuvent être remplacés par d'autres caractères. Ces caractères sont rangés dans la variable système #EDITCTL. Pour en changer, il faut modifier le contenu de cette variable, au moyen des directives MODIFY_PROFILE ou LET, avant la session MODIFY.

Il faut obligatoirement introduire trois caractères dans la variable #EDITCTL séparés par des espaces ou des virgules. Ils remplaceront les caractères '^', '\$' puis '[', dans cet ordre.

Attention : certains caractères peuvent donner des résultats inattendus comme, par exemple, le point d'interrogation (?) utilisé seul qui provoque la visualisation d'un texte HELP, ou la barre oblique (/) qui sera interprétée comme un caractère de séparation de paramètre. Il est donc recommandé de tester les caractères avant de rendre le changement "définitif" (c'est-à-dire de l'intégrer à une procédure de lancement).

Pour éviter toute ambiguïté, il est préférable de mettre les caractères choisis entre apostrophes.

Exemples :

```
MODIFY_PROFILE   EDITCTL = ('<', '>', '[');
                  changement des caractères ^ et $

LET   #EDITCTL ('{', '}', '<');
                  changement des trois caractères
```

3.11 TAMPON

Il est possible d'utiliser simultanément jusqu'à 19 tampons. Le tampon en cours de modification est toujours appelé tampon courant. Toutes les modifications portent, par défaut, sur ce tampon. Tous les autres sont des tampons auxiliaires.

Un tampon principal (MAIN) a déjà été créé lors du lancement de FSE et constitue le tampon courant, sauf s'il y a eu bascule de tampon dans la procédure de lancement (voir le paragraphe 3.16).

3.11.1 Nom de tampon

Un tampon est identifié par un nom de 8 caractères au maximum. Le premier caractère doit être une lettre (a à z), les suivants peuvent être des lettres, des chiffres (0 à 9) ou des caractères "souligné" (_).

Les noms de demande ne peuvent pas être utilisés pour des tampons.

Les tampons et leurs noms sont créés automatiquement la première fois qu'ils sont cités.

S'il n'y a aucune ambiguïté avec un autre nom de tampon (ou avec le nom d'une demande), il n'est pas nécessaire de fournir son nom complet pour désigner un tampon existant. L'intérêt du nom complet est néanmoins qu'il n'est jamais ambigu.

Exemple :

Soit les tampons BUF, BUFFER, LIST, et LINK :

BUF peut être désigné par BUF uniquement

BUFFER peut être désigné par BUFFER, BUFF ou BUF

LIST peut être désigné par LIST ou LIS

LINK peut être désigné par LINK ou LIN

3.11.2 Association d'un nom d'unité à un tampon.

A chaque tampon peut être associé un nom d'unité. Le nom d'unité associé est utilisé comme nom implicite au moment de la sauvegarde du contenu du tampon et également comme "mécanisme de protection" (voir le paragraphe 3.12).

Au moment où une unité est chargée dans le tampon, le nom de l'unité est associé au tampon. L'association d'un nom d'unité peut aussi être forcée, modifiée ou annulée au moyen de la demande NAME (voir la description de la demande NAME au chapitre 4). Le nom de l'unité associée s'affiche à l'écran (dans la zone en-tête). Sa visualisation peut également être obtenue au moyen de la demande DISP.

3.12 MECANISME DE PROTECTION

Un mécanisme de protection permet de gérer d'éventuels conflits dans le cas, par exemple, où plusieurs utilisateurs essaient de modifier la même unité au même moment.

Ce mécanisme est fourni dans deux situations différentes :

- (1) lors de la mise à jour de l'unité associée : si cette unité a été chargée à partir d'une bibliothèque LIB ou si le contenu du tampon a été précédemment sauvegardé dans l'unité associée. Le mécanisme de protection garantit alors :
 - que la version de l'unité utilisée pour les modifications est à jour au moment où ces modifications sont apportées (c'est-à-dire que l'unité n'a pas été modifiée depuis la dernière lecture ou sauvegarde). Si le contenu du tampon n'est pas à jour, il faut relire l'unité ou changer le nom associé.
 - qu'un autre utilisateur n'est pas en train de modifier la même unité au même moment. Si c'était le cas, il faudrait attendre que l'autre utilisateur ait fini de faire ses modifications avant de travailler sur cette unité. Mais il est également possible de décider que l'unité dans le tampon n'est pas la bonne et d'en changer.

L'unité qui fait l'objet de modifications est "réservée" à l'utilisateur jusqu'à la prochaine sauvegarde. D'autres utilisateurs peuvent cependant lire et visualiser cette unité mais sans pouvoir y apporter des modifications.

- (2) lors du chargement d'une nouvelle version de l'unité : si l'unité associée a été chargée à partir d'une bibliothèque INLIB ou si l'association du tampon a été forcée ou modifiée au moyen de la demande NAME. Le mécanisme de protection garantit alors que le contenu n'est pas modifié ailleurs.

Remarque : Sauf dans le premier cas (mise à jour de l'unité), il faudra donner au paramètre REPLACE la valeur 1, soit en utilisant la demande NAME soit lors de la sauvegarde.

FONCTIONS COMPLEXES

Les paragraphes suivants (jusqu'à 3.19 non compris) décrivent des fonctions plus complexes qui ne sont pas indispensables pour les utilisateurs débutants. Ceux-ci peuvent donc sauter la lecture de ces paragraphes.

3.13 FENETRE DE VISUALISATION

Pendant la modification et la visualisation du contenu du tampon courant, il est possible d'avoir à l'écran soit une autre partie de ce même tampon soit un autre tampon et ce, dans une autre fenêtre appelée "fenêtre de visualisation". La fenêtre utilisée pour l'édition, dans laquelle le contenu du tampon courant est visualisé, est appelée "fenêtre active".

Les fenêtres sont gérées au moyen des demandes "ligne" OW, CW, SB et TOP.

La fenêtre active peut se situer dans la partie supérieure ou inférieure de l'écran. Le positionnement de cette fenêtre est fait par FSE de façon à limiter autant que possible les déplacements de texte à l'écran.

La fenêtre active ne diffère de la fenêtre unique que par la taille de la zone texte. Neuf lignes de texte seulement sont visualisées dans la fenêtre active pour la grille principale ou la grille de guidage contre 20 lignes dans la fenêtre unique.

La fenêtre de visualisation est divisée en trois zones :

Zone	Lignes
Zone en-tête	1
Zone texte	2 à 9
Zone demande	10

Fenêtre de visualisation.

```

b=MAIN m=LIB: EXAMPLE DAT  > ln=23

 11 ===== text =====
 12 ===== text =====
 13 ===== text =====
 14 ===== text =====
 15 ===== text =====
15.1 ===== text =====
15.2 ===== text =====
15.3 ===== text =====

request _____ addr _____
    
```

3.13.1 Zone en-tête

b=<tampon> <unité> <lang> <dir> nbrel=<nlt>

Signification :

b=<tampon> :	nom du tampon
<unité> :	"no member" si aucune unité n'est associée au tampon, sinon : identificateur de bibliothèque et unité associée. (Voir le paragraphe 3.11.2).
<lang> :	= langage
<dir> :	= { > ! < }
nbrel=<nlt> :	= nombre de lignes dans le tampon

3.13.2 Zone texte

Cette zone permet d'afficher 9 lignes du tampon au maximum. Au moment de la visualisation du contenu du tampon, les lignes virtuelles TOP et BOTTOM se placent respectivement avant la première ligne et après la dernière ligne du tampon.

Chaque ligne de texte est précédée d'un numéro de ligne.

3.13.3 Zone demande (request)

Cette zone est utilisée pour les demandes relatives à la fenêtre de visualisation. Elle se compose d'une partie "nom de la demande" et d'une partie "paramètres", s'il y en a.

Les demandes qui concernent la fenêtre de visualisation sont les suivantes :

L	Localisation d'une ligne dont l'adresse est spécifiée. paramètre : une adresse de ligne spécifiée comme dans la partie adresse. Restriction : le caractère "=" n'est pas autorisé. Utiliser à la place la demande R.
>	saut d'une ou de plusieurs pages en avant. paramètre : nombre de pages. La valeur implicite est 1.
<	saut d'une ou plusieurs pages en arrière. paramètre : nombre de pages. La valeur implicite est 1.
R	Répétition de la dernière demande relative à la fenêtre de visualisation (L, < ou >). L'utilisateur est invité à répéter la demande jusqu'à l'effacement de la zone "request". paramètre : pas de paramètre
Q (ou /)	Sortie de la fenêtre de visualisation. (Cette demande est l'équivalent de la demande "ligne" CW). paramètre : pas de paramètre.

Les demandes concernant la fenêtre de visualisation sont exécutées après toutes les autres demandes relatives à la fenêtre active. Si une erreur est détectée pendant l'exécution d'une demande concernant la fenêtre de visualisation, le message approprié s'affiche à l'écran. L'utilisateur doit alors introduire la demande correcte, ou une barre oblique (/) dans la zone de gestion (ctl) qui a pour effet d'effacer la zone "request" et de mettre fin à l'exécution de la demande.

3.14 CONTEXTES

3.14.1 Identificateur de contexte

Si un ensemble de lignes (c'est-à-dire un contexte) doit être utilisé à plusieurs reprises, il est possible de lui affecter un "identificateur de contexte".

L'identificateur de contexte pourra être utilisé ensuite pour désigner les lignes en question (contexte).

Le calcul de l'adresse est fait une fois, au moment de l'affectation, ce qui signifie que l'affectation du contexte n'est pas changée par la renumérotation des lignes ou la modification de leur contenu.

Un identificateur de contexte est défini au départ au moyen d'une demande ASG et peut être modifié, redéfini, ou annulé au moyen des demandes "ligne" ADD, REM, ASG et CLC.

Les règles et conventions pour la citation et la définition de l'identificateur de contexte sont identiques à celles qui concernent les tampons mais elles s'appliquent au niveau du tampon; autrement dit, le même identificateur peut être utilisé pour désigner des contextes différents à condition que les tampons soient eux-aussi différents.

Une fois défini, l'identificateur de contexte peut être utilisé à la place d'une adresse, n'importe où dans une expression de demande "ligne". Certaines demandes ne pouvant porter sur tous les types d'adresse, un type est associé à l'identificateur de contexte. Les types d'identificateur de contexte sont les suivants : le type "ligne unique", le type "plage de lignes" et le type "ensemble de lignes" (comme pour les types d'adresse).

Le type d'identificateur de contexte est le type d'adresse utilisé au moment de l'affectation mais peut être changé au moyen des demandes "ligne" ADD et REM (voir la description de ces demandes au chapitre 4).

Pour plus de commodité, lorsqu'une plage ou un ensemble de lignes ne contiennent qu'une seule ligne, le type de contexte est changé en "ligne unique" et peut ainsi être utilisé par toutes les demandes.

Pour avoir la liste de tous les types d'identificateur de contexte, introduire DISP <nom_tampon>, ce paramètre indiquant le tampon pour lequel on souhaite avoir la liste des identificateurs de contexte.

Il est possible de vérifier à quoi correspond un identificateur de contexte au moyen de commandes telles que HL (affichage vidéo inverse) et UC (mise à jour de contexte).

Session Modify

Un contexte est identifié par un nom de 8 caractères au maximum. Le premier caractère doit être une lettre (a à z), les suivants peuvent être des lettres, des chiffres ou des caractères "souligné" (_).

Il n'est pas nécessaire de fournir le nom complet d'un identificateur, s'il n'y a aucun risque d'ambiguïté avec un autre identificateur du même tampon (ou un nom de demande). L'intérêt du nom complet est évidemment qu'il n'est jamais ambigu.

Exemple :

```
ASG */profil/ bloc;  
S   bl  /anc/ nouv/ ;  
ADD (4 8 */profil_2/) bl;
```

Dans cet exemple, l'identificateur de contexte 'bloc' est défini et désigné soit par 'bloc' soit par 'bl'.

Si un contexte correspondant à un identificateur devient vide, l'identificateur cesse d'être affecté.

Lorsqu'une plage de lignes est affectée à un identificateur, elle n'est définie que par les lignes début et fin. De ce fait, si de nouvelles lignes sont insérées dans une plage de lignes, elles lui sont intégrées. Ce n'est pas le cas pour un ensemble de lignes.

Exemple:

```
ASG *='ABC' ABC; ASG . TARGET; S ABC 'ABC' 'CDE'; KA ABC TARGET;
```

Dans tout le tampon, ABC est remplacé par CDE. Ensuite les lignes qui contenaient ABC sont copiées après la ligne courante précédente.

3.14.2 Mise à jour de contexte.

Le mode mise à jour de contexte permet l'édition d'un contexte ("ligne unique", "plage de lignes" ou "ensemble de lignes"), comme s'il s'agissait d'un tampon.

Les lignes situées hors du contexte ne sont ni visualisées, ni modifiées.

Le mode mise à jour de contexte est obtenu au moyen de la demande UC, la demande QC permettant d'en sortir.

En mode mise à jour, les grilles ne diffèrent que par la valeur de la partie "contexte" de la zone en-tête qui affiche soit l'identificateur de contexte soit l'adresse spécifiée au moyen de la demande UC. Si l'adresse est trop longue, elle apparaîtra tronquée dans la zone en-tête (toutes les informations de l'en-tête peuvent néanmoins être visualisées au moyen de la demande DISP).

Restrictions : Les demandes "ligne" suivantes ne sont pas autorisées en mode mise à jour de contexte :

- demandes portant sur des plages de lignes mais pas sur des ensembles de lignes :

SV, SVQ, C, CF, RB, IND

- demandes utilisées pour changer le tampon courant ou pour sortir de la session MODIFY :

SB, QUIT, Q, /

- demandes ayant une incidence sur l'affectation de l'identificateur de contexte :

ASG, REM, ADD, CLC, UC.

Lorsque de nouvelles lignes sont générées par les opérations de copie, d'insertion ou de lecture, ces lignes sont ajoutées au contexte courant (pas seulement au tampon). Cela signifie qu'au moment de la mise à jour d'un contexte spécifié par un identificateur, l'affectation correspondante est modifiée en conséquence.

Si le contexte mis à jour ne contient plus de lignes, il faut sortir du mode mise à jour de contexte puisqu'aucune ligne ne peut être désignée.

3.15 MACROS (EXECUTION DU CONTENU D'UN TAMPON)

Des séquences de demandes peuvent être enregistrées dans un tampon puis exécutées ultérieurement comme si elles provenaient de la zone "request" d'une grille principale. On peut ainsi définir des macros utilisateur.

L'exécution du contenu du tampon est demandée en spécifiant le nom du tampon, comme s'il s'agissait d'une demande, avec trois paramètres facultatifs.

Ainsi, le contenu du tampon 'upd_dcl' peut être exécuté en introduisant soit 'upd_dcl' soit 'upd_dcl par1 par2 par3;'.

Quand une demande provenant d'un tampon est exécutée et qu'un des paramètres (y compris l'adresse) est égal à [1, celui-ci est remplacé par le paramètre utilisé au moment de l'appel du tampon. De la même façon, [2 est remplacé par le second paramètre, [3 par le troisième, ce qui permet une passation de paramètres.

Exemple 1 :

La macro suivante permet de basculer sur un tampon spécifié, de l'effacer et d'y charger une unité d'une bibliothèque donnée (LIB étant la bibliothèque implicite).

Le contenu du tampon (appelé "load") est :

```
SB [1; CLB [1; RA . [2 [3;
```

Macro possible :

```
load doc doc_member;
```

[1 est remplacé par "doc", [2 par "doc_member" et [3 par une chaîne vide. Ainsi, les demandes suivantes pourront être exécutées :

```
SB doc; CLB doc; RA . doc_member, ;
```

Exemple 2 :

La macro suivante pourrait être utilisée en mode mise à jour de contexte, pour afficher dans la fenêtre de visualisation le texte qui précède et suit la ligne courante.

Le contenu du tampon (lw) est :

```
CW; OW; COM 'pour changer la ligne courante
de la fenêtre de visualisation'.
```

Macro possible :

```
UC contexte;OW;<liste de demandes>;lw;
```

Exemple 3 :

La macro suivante pourrait être utilisée pour visualiser le courrier en attente.

Le contenu du tampon ml est :

```
SEND mail;
```

Macro possible :

```
ml;
```

Exemple 4 :

La macro suivante peut être utilisée pour rechercher en avant une ligne contenant une chaîne de caractères donnée, cette chaîne figurant déjà dans la ligne courante.

Le contenu du tampon (lf) est :

```
L . 1; L [1;
```

Macro possible :

```
L /call /;
S . xx yy;
lf /call /;
```

[1 est remplacé par "/call /".

La macro est exécutée comme si la zone "request" avait été chargée ligne par ligne en mode expert ou demande par demande en mode novice.

La longueur d'une ligne de texte (d'un tampon) à exécuter ne doit pas dépasser 148 caractères. Tout comme pour une exécution normale, la grille peut être affichée avant l'exécution complète du contenu du tampon (par exemple, en cas d'erreur).

Dans ce cas, l'utilisateur peut modifier la demande, la ligne "request", le paramètre ou n'importe quel élément incorrect, comme pour une exécution normale.

L'exécution est ensuite reprise ou interrompue, selon la valeur indiquée dans la zone de gestion (ctl).

Session Modify

Remarque : Les demandes STP (Step - passage en mode pas à pas) et USTP (Unstep - fin du mode pas à pas) peuvent servir à tester une macro.

La ligne du tampon exécutée devient la ligne courante du tampon. Ainsi, au moment d'un test, il peut être utile d'ouvrir une fenêtre de visualisation au niveau du tampon d'exécution. Après la fin ou l'interruption de l'exécution, la ligne courante revient à sa valeur précédente. L'appel récursif des tampons est autorisé.

3.16 SEQUENCE DE LANCEMENT

Lorsque la session MODIFY est lancée, FSE recherche une unité dont le nom est "FSE_STARTUP_<nom utilisateur>" dans la bibliothèque LIB. Si elle est détectée, elle est chargée dans un tampon temporaire appelé STARTUP et exécutée.

En l'absence d'erreurs lors de l'exécution de la séquence de lancement, le tampon STARTUP est libéré et l'utilisateur est invité à introduire ses demandes.

Si une erreur est détectée, l'exécution est interrompue (comme c'est le cas lors de toute exécution du contenu d'un tampon). Le tampon STARTUP n'est pas libéré automatiquement et, s'il le désire, l'utilisateur peut relancer l'exécution.

Exemple d'utilisation :

FSE_STARTUP_MYNAME peut être constitué comme suit, pour charger, à chaque session d'édition, les trois macros décrites dans les exemples précédents :

```

SB load;
CLB load;
EA;-----début de la macro-load-----
SB [1; CLB [1; RA . [2 [3;
/
COM '-----fin de la macro LOAD-----'

SB lw;
CLB lw
EA; -----début de la macro lw -----
CW; OW;
/
COM '-----fin de la macro lw-----'

SB ml;
CLB ml
EA;-----début de la macro ml-----
SEND MAIL;
/
COM '-----fin de la macro ml-----'

COM '----toute autre définition de macro ou demande--'

SB MAIN;
```

Session Modify

Remarque : La demande CLB est utilisée de telle sorte qu'il est possible de réexécuter la séquence de lancement pendant la session. Les tampons contenant des macros sont réinitialisés, c'est-à-dire effacés avant d'être rechargés.

Exemple d'unité de lancement chargeant une macro appelée "add_mac" qui permet d'introduire d'autres macros dans l'unité de lancement elle-même.

```
SB add_mac;
CLB add_mac;
EA;-----début de add_mac-----
SB startup; CLB startup; RA . FSE_STARTUP_<idutil>;
EI ^; début d'insertion dans FSE_STARTUP_<idutil>
    SB <nom>;
    CLB <nom>;
    EA; -- >>>>-----début de <nom>-----
[[CC/
    COM '-----fin de <nom>-----';
[C/
COM '-----fin d'insertion dans FSE_STARTUP_<idutil>--';
S ^:. <nom> [1 ;
A ^/>>>>/; COM '--entrer texte de la nouvelle macro--';
SV;
D ^/<<<</ :$;
SB MAIN;
startup;
SB MAIN;
CLB startup;
/
COM '-----fin de add_mac-----';
SB MAIN;
```

Remarque : Il est conseillé de faire une copie de cet exemple et de s'en servir comme modèle. Il suffira de fournir un nom d'utilisateur à la place de <idutil> et de frapper la séquence d'échappement supplémentaire [C devant chaque séquence d'échappement.

Exemple d'utilisation de 'add_mac' :

```
add_mac save;
```

La macro 'add_mac' est lancée et il est demandé à l'écran d'introduire le contenu de la macro 'save'. Exemple :

```
RB; SV* [1 [2 [3;
```

Lorsque c'est fait, mettre un point-virgule (;) dans le zone de gestion (ctl) pour sortir du mode adjonction et pour mettre fin à l'exécution de la macro 'add_mac'.

Maintenant la macro 'save' est prête à être utilisée.

Exemple d'une unité de lancement qui en appelle une autre qui est située dans une bibliothèque donnée.

```
COM 'IL3 doit être affectée et my_startup doit';  
COM 'exister si l'unité de lancement existe dans LIB' ;  
SB lstart_b;  
RA . my_startup IL3;  
SB MAIN;  
start_b;  
SB MAIN;
```

Remarque : L'existence d'une unité de lancement est facultative.

3.17 TRAITEMENT DES CARACTERES NON GRAPHIQUES

FSE permet de traiter des caractères non graphiques :

- Si, dans la grille principale, un article contient des caractères non graphiques, ces caractères apparaissent sous forme de points. En mode écran (SE), les articles contenant des caractères non graphiques sont affichés en vidéo inverse et aucune mise à jour de ces articles n'est possible.
- Si la demande UL (mise à jour de ligne) est exécutée pour un article contenant des caractères non graphiques, ces caractères seront affichés sous la forme [Xhh (hh étant le code hexadécimal du caractère). Il est possible d'ajouter ou de changer des caractères non graphiques de l'article en utilisant la même grille.

En cas de mise à jour d'une ligne contenant les caractères '[', ces caractères prendront la forme '[c]', ceci afin de ne pas être interprétés avec leur signification spéciale.

A noter que la zone de mise à jour pour un article contenant des caractères non graphiques est agrandie et permet d'introduire jusqu'à 255 de ces caractères.

- Les demandes I (Insertion) et A (Adjonction) permettent également d'introduire de nouveaux articles contenant des caractères non graphiques qui sont codés sous la forme [Xhh.

Remarque : Le caractère '[' est le caractère CENT du profil. Si le texte contient trop de '[' à protéger, (en programmation Pascal, par exemple), il faudra changer le caractère CENT dans le profil.

S'il est nécessaire d'introduire des caractères '[' (dans les demandes UL, A ou I), il faudra les protéger en mettant '[c]' pour éviter qu'ils soient interprétés en tant que caractères spéciaux.

3.18 FONCTIONS D'AFFICHAGE, D'OCCULTATION ET DE MISE EN EVIDENCE

Les fonctions H (occultation) et HL (affichage en vidéo inverse) servent à mettre en évidence des parties du texte et sont faciles à utiliser (voir la description de ces demandes au chapitre 4).

3.19 DEMANDES DE MODIFY

En mode ligne, il y a 75 demandes utilisables. Elles sont présentées ci-dessous et regroupées par type de fonctions.

Une demande non formatée n'utilise aucune application (grille formatée) pour l'introduction ou la sortie des données, alors que les demandes formatées utilisent une application (grille formatée). Une application standard est fournie par le système pour chaque langage. Toutes les demandes formatées sont suivies de la lettre 'F'.

Demande d'aide (HELP)

? Affichage de la liste des demandes "ligne".

Demande de visualisation

DISP Affichage d'informations générales concernant les bibliothèques affectées, les tampons et les contextes.

Demandes d'adjonction/insertion/changement

A Adjonction de lignes non formatées, après une ligne spécifiée.
AF Adjonction de lignes formatées, après une ligne spécifiée.
I Insertion de lignes non formatées, avant une ligne spécifiée.
IF Insertion de lignes formatées, avant une ligne spécifiée.
C Remplacement d'une plage de lignes, par les lignes non formatées introduites.
CF Remplacement d'une plage de lignes, par les lignes formatées introduites.
EA Adjonction, après la ligne spécifiée, de la ligne du tampon d'exécution suivante. (Ne peut être appelée qu'à partir d'un tampon).
EI Insertion, avant la ligne spécifiée, de la ligne du tampon d'exécution suivante. (Ne peut être appelée qu'à partir d'un tampon).

Demandes de suppression

D Suppression d'un ensemble de lignes.
DQ Demande de confirmation de suppression, pour chaque ligne d'un ensemble donné et, en cas de réponse affirmative, suppression effective de la ligne.

Demandes de lecture

- RA Lecture d'une unité (ou d'une partie d'une unité) à partir d'une bibliothèque d'entrée et adjonction des lignes lues, après une ligne spécifiée.
- RI Lecture d'une unité (ou d'une partie d'une unité) à partir d'une bibliothèque d'entrée et insertion des lignes lues, avant une ligne spécifiée.

Demandes de copie/transfert

- KA Copie d'un ensemble de lignes et adjonction de ces lignes, un certain nombre de fois, après une ligne spécifiée.
- KI Copie d'un ensemble de lignes et insertion de ces lignes, un certain nombre de fois, avant une ligne spécifiée.
- MA Transfert d'un ensemble de lignes et adjonction de ces lignes, un certain nombre de fois, après une ligne spécifiée.
- MI Transfert d'un ensemble de lignes et insertion de ces lignes, un certain nombre de fois, avant une ligne spécifiée.
- K Copie d'un ensemble de lignes et adjonction des lignes copiées, après la ligne courante d'un tampon auxiliaire spécifié.
- KR Copie d'un ensemble de lignes et remplacement du contenu d'un tampon auxiliaire spécifié, par les lignes copiées.
- M Transfert d'un ensemble de lignes et adjonction des lignes transférées, après la ligne courante d'un tampon auxiliaire spécifié.
- MR Transfert d'un ensemble de lignes et remplacement du contenu d'un tampon auxiliaire spécifié, par les lignes transférées.
- IB Insertion, un certain nombre de fois, des lignes d'un tampon auxiliaire spécifié, avant chaque ligne d'un ensemble de lignes.
- AB Adjonction, un certain nombre de fois, des lignes d'un tampon auxiliaire spécifié, après chaque ligne d'un ensemble de lignes.

Demandes de modification

- S Remplacement de zéro, une ou plusieurs occurrences d'une chaîne de caractères donnée, par une nouvelle chaîne et ce, sur chaque ligne d'un ensemble de lignes.
- SQ Demande, pour chaque ligne d'un ensemble de lignes contenant au moins une occurrence d'une chaîne de caractères donnée, si la chaîne de caractères doit être remplacée par une nouvelle chaîne.
- UL Mise à jour, ligne par ligne, d'un ensemble de lignes non formatées.
- ULF Mise à jour, ligne par ligne, d'un ensemble de lignes formatées.

Session Modify

LO	Conversion des caractères majuscules en minuscules pour chaque ligne d'un ensemble de lignes.
UP	Conversion des caractères minuscules en majuscules pour chaque ligne d'un ensemble de lignes.

Demandes d'instauration/sortie de mode

SE	Passage en mode écran.
LE	Passage en mode ligne.
UC	Passage en mode mise à jour de contexte.
QC	Sortie du mode mise à jour de contexte.
EXP	Passage en mode expert.
NOV	Passage en mode novice.

Demandes de positionnement

L	Localisation d'une nouvelle ligne courante et positionnement de cette ligne au centre de la zone texte.
>	Déplacement de la fenêtre d'un (ou plusieurs) écran(s) vers le bas.
<	Déplacement de la fenêtre d'un (ou plusieurs) écran(s) vers le haut.

Demandes d'affichage/mise en évidence/occultation

PF	Affichage, avec un format spécifié, de chaque ligne d'un ensemble de lignes.
HL	Affichage en vidéo inverse de chaque ligne d'un ensemble de lignes. (Les grilles ne contenant aucune des lignes de l'ensemble sont sautées).
H	Occultation de toutes les lignes d'un ensemble pour qu'elles ne soient pas affichées à l'écran.
UH	Suppression de l'occultation des lignes précédemment cachées.

Demande de sortie de session MODIFY

QUIT	Sortie de la session MODIFY.
Q	Sortie de la session MODIFY.
/	Sortie de la session MODIFY.

Demande de renumérotation

RB	Renumerotation d'une plage de lignes se trouvant dans le tampon.
----	--

Demandes de sauvegarde

NAME	Changement du nom de l'unité associée et de tous les autres paramètres de sauvegarde (langage, longueur d'article, statut REPLACE).
SV	Sauvegarde du contenu du tampon dans la bibliothèque de sortie.
SVQ	Sauvegarde du contenu du tampon dans la bibliothèque de sortie et sortie de la session MODIFY.

Demandes de spécification d'option de saut

FC	Spécification d'une option de saut pour chaque ligne d'un ensemble de lignes.
----	---

Demande de concaténation/coupure de ligne

SP	Coupure de chaque ligne d'un ensemble de lignes avant chaque occurrence d'une chaîne de caractères donnée.
CC	Concaténation de chaque ligne d'un ensemble de lignes avec la ligne précédente.

Demandes de gestion des identificateurs de contexte.

ASG	Affectation d'un identificateur de contexte à un ensemble de lignes, une plage de lignes ou une seule ligne.
ADD	Adjonction d'un ensemble de lignes à un contexte déjà affecté.
REM	Retrait d'un ensemble de lignes d'un contexte déjà affecté.
CLC	Effacement de l'identificateur de contexte.

Demandes relatives au tampon

SB	Basculement sur un tampon spécifié.
CLB	Effacement de tampon.

Demande d'exécution du contenu du tampon

<nom du tampon>	Exécution du contenu d'un tampon donné.
-----------------	---

Demande de spécification de commentaire

COM	Insertion d'un commentaire
-----	----------------------------

Demandes d'affectation de bibliothèque

LIB	Association d'une bibliothèque à l'identificateur de bibliothèque de sortie LIB.
IL1	Association d'une bibliothèque à l'identificateur de bibliothèque de sortie IL1.
IL2	Association d'une bibliothèque à l'identificateur de bibliothèque de sortie IL2.
IL3	Association d'une bibliothèque à l'identificateur de bibliothèque de sortie IL3.

Demandes de gestion de fenêtre

OW	Ouverture de la fenêtre de visualisation (si celle-ci ne l'est pas) et affichage dans cette fenêtre du contenu d'un tampon auxiliaire (ou du tampon courant).
CW	Fermeture de la fenêtre de visualisation.
TOP	Positionnement de la fenêtre active en haut de l'écran (n'a d'incidence que sur la présentation).

Demande de répétition

R	Répétition de la dernière demande "ligne" exécutée.
=	Visualisation, dans la zone "request", de la dernière demande "ligne" exécutée (à l'exception des demandes R et =).

Demande relative aux bornes

BNDS	Définition (ou suppression) de bornes.
------	--

Demande d'utilisation des retraits

IND	Utilisation des retraits pour la procédure GPL
-----	--

Demande d'exécution de commandes GCL

SEND	Envoi des commandes GCL au système.
------	-------------------------------------

Demandes relatives au mode pas à pas

STP	Passage en mode pas à pas, qui provoque l'affichage de chaque demande avant son exécution.
USTP	Sortie du mode pas à pas.

4. Demandes "LIGNE"

Ce chapitre fournit une description de chaque demande "ligne" utilisable dans le cadre de la session MODIFY.

A noter que le terme "demandes" s'applique aux fonctions de modification et le terme "commandes" aux fonctions de FSE appelées au niveau commande (caractères de guidage F:).

La demandes sont décrites en ordre alphabétique, chaque demande commençant sur une nouvelle page.

Pour chaque demande, on trouvera sa fonction, la signification des paramètres et quelques exemples. Pour certaines demandes (L, D et S), les exemples sont plus nombreux. Il est recommandé aux lecteurs qui préfèrent les exemples aux explications abstraites de se reporter à la description de ces demandes.

4.1 DEMANDE A

La demande A (Append) permet d'ajouter des lignes non formatées après une ligne spécifiée.

Paramètres de la demande :

paramètre adresse : ligne unique
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Le paramètre adresse spécifie la ligne après laquelle les lignes doivent être ajoutées. La demande A provoque le passage en mode saisie.

Pour plus de détails, voir le paragraphe 3.8 qui traite du mode saisie.

Après exécution de la demande la ligne courante est la dernière ligne introduite.

Remarque importante :

Pour sortir du mode saisie, placer le curseur dans la zone de gestion (ctl) de la grille (en appuyant sur la touche ORIGINE - HOME). Introduire ensuite un point-virgule (;).

Exemples :

a

Passage en mode saisie, l'introduction commençant après la ligne courante.

a 640

Passage en mode saisie, l'introduction commençant après la ligne 640.

a \$

Passage en mode saisie, l'introduction commençant après la dernière ligne du tampon.

4.2 DEMANDE AB

La demande AB (Append Buffer) ajoute le contenu d'un tampon auxiliaire après chaque ligne d'un ensemble de lignes spécifié et ce, un certain nombre de fois.

Paramètres de la demande :

paramètre adresse	:	ensemble de lignes
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	
premier paramètre	:	tampon
valeur implicite	:	aucune (ce paramètre est obligatoire)
deuxième paramètre	:	facteur de répétition
valeur implicite	:	1

Le premier paramètre supplémentaire spécifie le tampon auxiliaire à partir duquel les lignes doivent être copiées.

Le deuxième paramètre est un facteur de répétition et doit être un entier positif.

Après exécution de la demande, la ligne courante est la dernière ligne ajoutée.

Exemples :

```
ab 10 DATA
```

Le contenu du tampon DATA est ajouté après la ligne 10.

```
ab 1:12 DATA
```

Le contenu du tampon DATA est ajouté après chacune des lignes 1 à 12.

```
ab 10:120 DATA 4
```

Quatre copies du contenu du tampon DATA sont ajoutées après chacune des lignes 10 à 120.

4.3 DEMANDE ADD

La demande ADD (ADD context) ajoute un ensemble de lignes à un contexte désigné par un identificateur précédemment affecté. Au cours de cette opération, seules les lignes n'appartenant pas encore au contexte sont ajoutées.

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires :
  premier paramètre       : contexte
  valeur implicite        : aucune (ce paramètre est obligatoire)
```

Le premier paramètre supplémentaire spécifie l'identificateur de contexte (voir le paragraphe "Identificateur de contexte" au chapitre 3).

L'identificateur doit déjà exister (c'est-à-dire avoir été affecté).

Après exécution de la demande ADD, le "type" de l'identificateur de contexte est "ensemble de lignes" sauf si l'identificateur du contexte résultant ne désigne qu'une "seule ligne", auquel cas le type est "ligne unique".

Après exécution de la demande, la ligne courante est inchangée.

Exemples :

```
add . BLOC
```

La ligne courante est ajoutée à l'ensemble de lignes désigné par l'identificateur de contexte BLOC.

```
add */BEGIN/ BLOC
```

Toutes les lignes contenant la chaîne de caractères BEGIN sont ajoutées à l'ensemble de lignes désigné par l'identificateur de contexte BLOC.

```
add FINBLOC BLOC
```

L'ensemble de lignes spécifié par l'identificateur de contexte FINBLOC est ajouté à l'ensemble de lignes désigné par l'identificateur de contexte BLOC.

4.4 DEMANDE AF

La demande AF (Append Formatted) permet d'ajouter des lignes formatées après une ligne spécifiée.

Paramètres de la demande :

paramètre adresse : ligne unique
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Cette demande appelle l'application, fournie par le système, qui correspond au langage utilisé, pour introduire de nouvelles lignes formatées. Quand les lignes à ajouter ont été introduites, il faut mettre un caractère barre oblique (/) dans la zone EXIT pour sortir du mode saisie formatée.

Les caractères non graphiques ne peuvent pas être utilisés avec cette demande; si le texte à introduire en contient, il faut utiliser la demande A.

Après exécution de la demande, la ligne courante est la dernière ligne introduite ou, si aucune ligne n'a été introduite, la valeur du paramètre adresse.

Les formats standards des langages sont acceptés (voir le paragraphe 3.9).

Si une ligne plus longue que ne l'autorise le format standard correspondant est introduite, elle sera tronquée.

Exemples :

af 640

Passage en mode saisie formatée, l'introduction commençant après la ligne 640.

af 12.3

Passage en mode saisie formatée, l'introduction commençant après la ligne 12.3.

af ^

Passage en mode saisie formatée, l'introduction commençant après la première ligne du tampon.

af \$

Passage en mode saisie formatée, l'introduction commençant après la dernière ligne du tampon.

4.5 DEMANDE ASG

La demande ASG (ASSiGn context identifier) permet d'affecter un identificateur de contexte à un ensemble de lignes spécifié.

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
valeur implicite           : la ligne courante
paramètres supplémentaires :
premier paramètre          : contexte
valeur implicite           : aucune (ce paramètre est obligatoire)
```

Le premier paramètre supplémentaire spécifie l'identificateur de contexte à affecter.

S'il y a déjà un identificateur de contexte pour le tampon courant, il est ré-affecté, sinon il est créé automatiquement (voir le paragraphe 3.14.1).

Après exécution de la demande, la ligne courante est inchangée.

Exemples :

```
asg 20:30 BLOC
```

L'identificateur de contexte BLOC est affecté à l'ensemble de lignes 20 à 30.

```
asg (1,7,10) BLOC
```

L'identificateur de contexte BLOC est affecté à l'ensemble constitué par les lignes 1, 7 et 10.

```
asg */PROC/ BLOC
```

L'identificateur de contexte BLOC est affecté à l'ensemble des lignes contenant la chaîne de caractères PROC.

```
asg *=[^/END/ NOTEND
```

L'identificateur de contexte NOTEND est affecté à l'ensemble des lignes contenant la chaîne de caractères END.

4.6 DEMANDE BNDS

La demande BNDS (BouNDS) définit une plage de texte verticale dans laquelle la recherche de chaîne régulière suivante doit être faite. La valeur implicite est la totalité du texte et peut être annulée et remplacée par une autre au moyen de la même demande.

Attention : La mise en place de "bornes" délimitant une plage de texte s'effectue simultanément sur tous les tampons à la fois et reste active jusqu'à son changement explicite. Elle s'applique à n'importe quelle recherche de chaîne régulière (dans le paramètre adresse et dans les paramètres "chaînes" des demandes S et SP). Lorsque les bornes ne correspondent pas à la totalité du texte (valeur implicite), elles sont affichées dans la zone "en-tête".

Paramètres de la demande :

```
paramètre adresse           : aucun
paramètres supplémentaires :
  premier paramètre        : colonne gauche
  valeur implicite         : 1
  deuxième paramètre       : colonne droite
  valeur implicite         : 255
```

Les premier et deuxième paramètres supplémentaires spécifient les colonnes de gauche et de droite qui délimitent verticalement la plage de texte.

Ces paramètres doivent être des nombres compris entre 1 et 255 inclus, le premier paramètre étant nécessairement inférieur ou égal au second paramètre.

Il faut spécifier soit zéro paramètre soit les deux paramètres.

L'absence de paramètre signifie que les bornes doivent avoir leur valeur implicite, soit 1 et 255.

Exemples :

```
bnds 29 31
```

La plage est délimitée par les colonnes 29 et 31 incluses.

```
bnds
```

La plage s'étend de la colonne 1 à la colonne 255 (l'une et l'autre étant incluses).

4.7 DEMANDE C

La demande C (Change) permet de supprimer une plage de lignes non formatées et de la remplacer par des lignes introduites en mode saisie.

Paramètres de la demande :

paramètre adresse : ligne unique ou plage de lignes
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Cette demande provoque le passage en mode saisie (voir le paragraphe 3.8 qui traite de ce mode).

Après exécution de la demande, la ligne courante est la dernière ligne introduite. Si aucune ligne n'a été introduite, la ligne courante est la première ligne après la plage de lignes spécifiée; si cette plage de lignes contient la dernière ligne de l'unité, la nouvelle dernière ligne devient la ligne courante.

Remarque importante :

Pour sortir du mode saisie, placer le curseur dans la zone "ctl" de la grille (en appuyant sur la touche ORIGINE - HOME) puis introduire un point-virgule (;).

Exemples :

c

Passage en mode saisie, l'introduction commençant au niveau de la ligne courante qui est supprimée.

c 640

Passage en mode saisie, l'introduction commençant au niveau de la ligne 640 qui est supprimée.

4.8 DEMANDE CC

La demande CC (ConCatenate) permet de concaténer chaque ligne d'un ensemble de lignes spécifié avec la ligne qui précède. Pour la première ligne, cette opération n'est évidemment pas possible.

Paramètres de la demande :

paramètre adresse : ensemble de lignes
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est la dernière ligne de l'ensemble.

Exemple :

```
cc 0.3
```

La ligne 0.3 est concaténée avec la ligne 0.2.

Soit un tampon contenant les lignes suivantes :

```
1 AAA  
2 BB  
3 CCC  
4 D  
5 EE  
6 FFFF
```

L'exécution de la demande

```
CC (2:3 5);
```

donnera :

```
1 AAABBCCC  
4 DEE  
6 FFFF
```

4.9 DEMANDE CF

La demande CF (Change Formatted) permet de supprimer une plage de lignes et de la remplacer par des lignes formatées introduites en mode saisie.

Paramètres de la demande :

paramètre adresse : ligne unique ou plage de lignes
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Cette demande appelle une application, fournie par le système et qui correspond au langage utilisé, pour introduire les nouvelles lignes en mode formaté.

Les lignes spécifiées par le paramètre adresse sont supprimées.

Après avoir introduit les lignes destinées à remplacer la plage de lignes spécifiée, mettre une barre oblique (/) dans la zone EXIT pour sortir du mode saisie formatée.

Les caractères non graphiques ne peuvent pas être utilisés avec cette demande; si le texte à introduire en contient, il faut utiliser la demande C.

Les formats standards des langages sont acceptés (voir au chapitre 3 le paragraphe concernant le mode formaté). Si une ligne plus longue que ne l'autorise le format standard correspondant est introduite, elle sera tronquée.

Remarque importante :

Pour sortir du mode saisie formatée, placer le curseur dans la zone "ctl" de la grille (en appuyant sur la touche ORIGINE - HOME), puis introduire une barre oblique (/).

Exemple :

cf 640

Passage en mode saisie formatée, l'introduction commençant au niveau de la ligne 640, qui est supprimée.

4.10 DEMANDE CLB

La demande CLB (CLear Buffer) permet d'effacer le contenu d'un tampon spécifié et de le libérer, à condition qu'il ne s'agisse pas du tampon courant ou du tampon "visualisé" :

```
paramètre adresse           : aucun
paramètres supplémentaires :
  premier paramètre        : tampon
  valeur implicite         : aucune (ce paramètre est
                             obligatoire)
```

Le premier paramètre supplémentaire spécifie le nom du tampon à effacer. (Voir au chapitre 3 le paragraphe "Nom de tampon").

Le tampon spécifié doit exister. D'autre part, un tampon ne peut pas être libéré lorsque son contenu est en cours d'exécution.

Remarque : Dans les exemples suivants, les demandes sont exécutées à partir du tampon principal (main).

Exemples :

```
clb main
```

Le tampon principal (main) est effacé (autrement dit, son contenu est supprimé). A noter qu'il est plus efficace d'utiliser cette demande que la demande de suppression correspondante (d *) et que le tampon "main" n'est pas libéré puisqu'il est en fait le tampon de travail.

```
clb DRAW_BOX
```

Le tampon DRAW_BOX est effacé ET libéré (puisque'il n'est pas le tampon de travail). L'exécution de cette demande est nécessaire lorsque, par exemple, le nombre de tampons affectés a atteint le maximum autorisé (c'est-à-dire 18 tampons auxiliaires).

4.11 DEMANDE CLC

La demande CLC (CLear Context) permet de libérer un identificateur de contexte.

Paramètres de la demande :

```
paramètre adresse           : aucun
paramètres supplémentaires :
  premier paramètre        : contexte ou *
  valeur implicite         : aucune (ce paramètre est
                             obligatoire)
```

Le premier paramètre supplémentaire spécifie l'identificateur de contexte. (Voir au chapitre 3 le paragraphe "Identificateur de contexte").

Cet identificateur doit déjà exister (c'est-à-dire avoir été affecté) pour le tampon courant.

CLC *; signifie que tous les identificateurs de contexte du tampon courant doivent être libérés.

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

```
clc BLOC
```

L'identificateur de contexte BLOC est libéré.

4.12 DEMANDE COM

La demande COM (COMment) permet d'insérer des commentaires. Elle s'utilise pour un tampon dont le contenu doit être exécuté; sinon, elle est acceptée mais inutile.

Paramètres de la demande :

```
paramètre adresse           : aucun
paramètres supplémentaires :
  premier paramètre         : commentaire
  valeur implicite          : aucune (ce paramètre est
                             obligatoire)
```

Le premier paramètre supplémentaire est le commentaire lui-même.

Un commentaire est une chaîne de caractères délimitée par deux barres obliques (/) ou par deux apostrophes (').

Les caractères "/" ou " ' " peuvent être insérés dans le commentaire en les faisant précéder de la séquence d'échappement [C.

S'il n'y a aucun séparateur (", " " ";" ou "(" en début de chaîne) ou délimiteur (barre oblique ou apostrophe) à l'intérieur du commentaire, il est possible d'omettre les délimiteurs en début et fin de commentaire.

Exemple :

```
com 'la macro EXJ exécute un travail.'
```

Cette demande est utilisée pour annoter les commandes dans un tampon d'exécution.

4.13 DEMANDE CW

La demande CW (Close Window) permet de fermer la fenêtre de visualisation, si celle-ci est ouverte.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

La ligne courante n'est pas changée.

Exemple :

CW

La fenêtre de visualisation ouverte est fermée.

4.14 DEMANDE D

La demande D (Delete) permet de supprimer un ensemble de lignes spécifié.

Paramètres de la demande :

paramètre adresse : ensemble de lignes
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est la ligne qui suit la dernière ligne supprimée. Si la dernière ligne du tampon est supprimée, c'est la nouvelle dernière ligne qui devient la ligne courante.

Exemples :

```
d
```

La ligne courante est supprimée.

```
d 20
```

La ligne 20 est supprimée.

```
d 610:660
```

Les lignes 610 à 660 sont supprimées.

```
d 325.10:325.18
```

Les lignes 325.10 à 325.18 sont supprimées.

```
d */vingt/
```

Toutes les lignes du tampon contenant "vingt" sont supprimées. Noter que l'astérisque (*) est l'équivalent de "^:\$" mais il est plus rapide d'utiliser l'astérisque.

```
d (1,4,13 8:11)
```

Les lignes 1, 4 et 13 ainsi que les lignes 8 à 11 sont supprimées.

```
d 325.10:325.18=/quarante/
```

Toutes les lignes de 325.10 à 325.18 qui contiennent la chaîne "quarante" sont supprimées.

```
d *=[^/quarante/
```

FSE - Editeur Plein Ecran

Toutes les lignes du tampon qui ne contiennent pas la chaîne "quarante" sont supprimées.

```
d */^quarante/
```

Toutes les lignes du tampon qui commencent par la chaîne "quarante" sont supprimées.

```
d 325.10:325.18=/quarante/[^cinquante/
```

Toutes les lignes de 325.10 à 325.18 qui contiennent la chaîne "quarante" et la chaîne "cinquante" sont supprimées.

```
d 325.10:325.18=/quarante/[!/cinquante/
```

Toutes les lignes de 325.10 à 325.18 qui contiennent la chaîne "quarante" ou la chaîne "cinquante" sont supprimées.

```
d OLD_EXAM
```

Toutes les lignes spécifiées au moyen de l'identificateur de contexte OLD_EXAM sont supprimées.

4.15 DEMANDE DISP

La demande DISP (DISPlay) permet d'afficher deux types de grilles. Le première fournit des informations d'ordre général sur le mode de FSE en cours ainsi que sur les bibliothèques affectées et le nom et l'état du (ou des) tampon(s) utilisé(s). Le second type de grilles fournit des informations détaillées sur chaque tampon.

Si un nom de tampon est spécifié en tant que premier paramètre de la demande, les informations détaillées affichées sont celles qui concernent ce tampon uniquement. Appuyer sur la touche de transmission (TRANS) pour sortir de DISP.

Si aucun nom de tampon n'est spécifié, la grille qui fournit les informations générales s'affiche en premier, après quoi il est possible de faire défiler à l'écran les grilles donnant des informations détaillées sur chaque tampon et ce, jusqu'à ce qu'une barre oblique (/) soit introduite dans la zone de gestion (ctl) afin de sortir de DISP.

L'introduction des signes '>' et '<' dans la zone de gestion (ctl) permet d'afficher la grille d'état détaillée du tampon qui suit ou précède respectivement le tampon courant.

Remarque : Si "?GR4" est spécifié à la place du nom du tampon, c'est la grille d'informations générales qui s'affiche (comme si le nom de tampon avait été purement et simplement omis) ainsi que le contenu du registre GR4. Ces informations peuvent utiles en cas de problèmes d'E/S ou d'affectation de bibliothèques.

Paramètres de la demande :

paramètre nom : nom du tampon
valeur implicite : informations générales
paramètres supplémentaires : aucun

Exemples :

```
disp
```

Affichage d'informations générales.

```
disp MAIN
```

Affichage d'informations détaillées concernant le tampon MAIN.

FSE - Editeur Plein Ecran

Exemple de grilles obtenues au moyen de la demande DISP.

```
GENERAL INFORMATION                                ctl  _____

mode :      line editor
           expert

libraries :
•           LIB      = LMN. DOC
           INLIB1   = LMN. FSE. SLLIB
           INLIB2   not assigned
           INLIB3   not assigned
used buffer(s) :
           MAIN      NEXT      ->REQ  DOC (m)
                                           -

(m) : modified buffer -> : current buffer
```

```
buffer MAIN                                       ctl  _____

member      = DOC_FSE_V3
input library = LMN. DOC
language    = DAT
no record length
modified    = NO
replace     = YES
no bounds
line count  = 4030
no context in use
```

Demandes "Ligne"

```
buffer NEXT —— ctl  
  
input library      = LMN. DOC  
language           = DAT  
record length     = 50  
modified          = NO  
replace           = NO  
no bounds  
line count        = 7  
no context in use
```

```
buffer REQ_DOC —— ctl  
  
member            = N  
member            = REQUEST_DOC  
input library     = LMN. DOC  
language          = DAT  
no record length  = YES  
modified          = YES  
replace           = 5: 600=/string/  
context mode  
no bounds         = 1240  
line count  
context(s) in use :  
    REQNAME (s) REC_ LN (s) EX1 (r) HEADER (l)
```

(s) : set , (l) : line , (r) : range.

4.16 DEMANDE DQ

La demande DQ (Delete Query) permet de supprimer, après confirmation de l'utilisateur, un ensemble de lignes spécifié.

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires : aucun
```

FSE demande, pour chacune des lignes de l'ensemble spécifié, ce qu'il doit faire. A cet effet, une grille contenant la ligne susceptible d'être supprimée et le message de guidage suivant s'affiche :

```
'TRANSMIT TO CONTINUE, "/" TO QUIT, "S" TO SKIP, "L" TO LEAVE QUERY MODE'
```

("Appuyer sur TRANS pour continuer, "/" pour sortir de DQ, "S" pour sauter cette ligne, "L" pour mettre fin au mode demande de confirmation")

- Quand l'utilisateur se contente d'appuyer sur la touche de transmission (TRANS), c'est-à-dire lorsqu'il n'introduit aucune des valeurs autorisées dans la zone de gestion (ctl), la ligne courante est supprimée et l'exécution se poursuit normalement, la ligne suivante susceptible d'être effacée étant affichée ainsi que le message de guidage décrit ci-dessus.
- Quand l'utilisateur introduit l'une des valeurs suivantes dans la zone de gestion (ctl), l'opération exécutée est fonction de cette valeur :

```
"/" : la ligne n'est pas supprimée et il est mis fin à la demande de
      suppression
"S" : la ligne n'est pas supprimée et l'exécution se poursuit
"L" : la ligne est supprimée, et les suppressions suivantes sont
      effectuées sans demande préalable de confirmation.
```

Exemple :

```
dq *=/Jean/
```

La première occurrence de la chaîne de caractères "Jean" est recherchée. Lorsqu'elle est détectée, FSE affiche le message de guidage décrit ci-dessus. En cas d'appui sur la touche de transmission (TRANS), la ligne est supprimée. Si "S" est introduit, la ligne n'est pas supprimée. Dans les deux cas, la prochaine occurrence de la chaîne de caractères "Jean" est recherchée et, si elle est détectée, le message de guidage est visualisé.

4.17 DEMANDE EA

La demande EA (Enter Append) ne peut être utilisée qu'à partir d'un tampon (tampon d'exécution, par exemple). Les lignes qui suivent cette demande dans un tampon d'exécution sont considérées comme le texte d'entrée jusqu'à ce que la fin du tampon soit atteinte ou la ligne "fin d'entrée" détectée. Ce texte est ajouté, après la ligne spécifiée, dans le tampon courant.

Paramètres de la demande :

```
paramètre adresse          : ligne unique
    valeur implicite       : la ligne courante
paramètres supplémentaires : aucun
```

La ligne "fin d'entrée" est une ligne qui commence par une barre oblique (/) suivie de n'importe quel nombre d'espaces (y compris zéro).

Remarque : Si la demande EA se termine par un point-virgule (;) le reste de la ligne est sauté.

Des caractères non graphiques peuvent être introduits sous la forme [Xhh, hh étant le code hexadécimal du caractère.

La signification spéciale des caractères : "[" "x" "/" et "c" peut être annulée au moyen de la séquence d'échappement [c.

Ainsi, on utilisera [c/ pour obtenir / dans une ligne et [[cc pour obtenir [c.

Après exécution de la demande, la ligne courante est la dernière ligne introduite.

Exemple :

```
EA 5; ce texte (après le point-virgule) est ignoré.
première ligne à ajouter
deuxième ligne à ajouter
dernière ligne à ajouter
/
```

4.18 DEMANDE EI

La demande EI (Enter Insert) ne peut être utilisée qu'à partir d'un tampon (tampon d'exécution, par exemple). Les lignes qui suivent cette demande dans un tampon d'exécution sont considérées comme le texte d'entrée jusqu'à ce que la fin du tampon soit atteinte ou la ligne "fin d'entrée" détectée. Ce texte est inséré, avant la ligne spécifiée, dans le tampon courant.

Paramètres de la demande :

```
paramètre adresse          : ligne unique
    valeur implicite       : la ligne courante
paramètres supplémentaires : aucun
```

La ligne "fin d'entrée" est une ligne qui commence par une barre oblique (/) suivie de n'importe quel nombre d'espaces (y compris zéro).

Remarque : Si la demande EI se termine par un point-virgule (;) le reste de la ligne est sauté.

Des caractères non graphiques peuvent être introduits sous la forme [Xhh, hh étant le code hexadécimal du caractère.

La signification spéciale des caractères : "[" "x" "/" et "c" peut être annulée au moyen de la séquence d'échappement [c.

Ainsi, on utilisera [c/ pour obtenir / dans une ligne et [[cc pour obtenir [c.

Après exécution de la demande, la ligne courante est la dernière ligne introduite.

Exemple :

```
EI 5; ce texte (après le point-virgule) est ignoré.
première ligne à insérer
deuxième ligne à insérer
dernière ligne à insérer
/
```

4.19 DEMANDE EXP

La demande EXP (EXPert) permet de choisir le mode expert. Ce mode est implicite pour des utilisateurs expérimentés (c'est-à-dire des utilisateurs dont la variable GCL #NOVICE a la valeur 0).

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

```
exp
```

Passage en mode expert.

Remarque : Pour sortir du mode expert, introduire "nov".

4.20 DEMANDE FC

La demande FC (Form Control) permet de forcer un caractère de commande de saut dans l'en-tête SSF d'un ensemble de lignes.

Paramètres de la demande :

```
paramètre adresse      : ensemble de lignes
  valeur implicite     : la ligne courante
paramètre supplémentaire : caractère de commande de saut
  valeur implicite     : espace
```

Le paramètre supplémentaire est le caractère de commande de saut qui peut être l'un des suivants :

Caractère	Signification
espace	interligne normal
0	saut d'un ligne blanche avant la ligne considérée
-	saut de deux lignes blanches avant la ligne considérée
+	recouvrement de la ligne précédente
1	saut en haut de page avant la ligne considérée

Après exécution de la demande, la ligne courante est la dernière ligne de l'ensemble de lignes spécifié.

Le caractère de commande de saut peut être protégé par deux barres obliques (/) ou deux apostrophes (').

Le caractère de commande de saut + doit obligatoirement être protégé du fait qu'il a une signification particulière au niveau de la fonction grille "grande zone".

Exemples :

```
FC *='PROC' '1' ;
```

Toutes les lignes contenant la chaîne de caractères 'PROC' auront le caractère de commande de saut '1', ce qui les placera en haut de page.

```
FC (1 5) '+' ;
```

Le caractère de commande de saut pour les lignes 1 et 5 étant +, il y aura, à chaque fois, recouvrement de la ligne précédente.

```
FC 8 ;
```

En ce qui concerne la ligne 8, le caractère de commande de saut est la valeur implicite, c'est-à-dire l'espace qui correspond à un interligne normal.

4.21 DEMANDE H

La demande H (Hide) permet de cacher certaines parties du texte afin :

- de mettre en valeur d'autres parties de ce texte,
- d'obtenir sur un même écran, des parties de texte qui autrement n'auraient pas pu être visualisées sur le même écran du fait qu'elles dépassent les 20 lignes.

Les parties du texte qui sont cachées sont signalées par une ligne spéciale.

A noter que les lignes cachées restent visibles pour FSE et que, par conséquent, elles sont traitées lorsqu'elles sont à l'intérieur d'une plage de lignes citée dans une demande.

Paramètres de la demande :

```
paramètre adresse          : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires : aucun
```

Exemples :

```
h 70:300
```

Les lignes 70 à 300 sont cachées.

```
h *=[^/Exemple/
```

Toutes les lignes du tampon qui ne contiennent PAS la chaîne de caractères "Exemple" sont cachées.

Exemple d'utilisation de la demande H : h 170:890

Avant l'exécution de la demande, la grille se présente comme suit :

```
b=MAIN m=LIB: DEM DAT > ln=2470                                ctl ____  
  100 this is record no 100  
  110 this is record no 110  
  120 this is record no 120  
  130 this is record no 130  
  140 this is record no 140  
  150 this is record no 150  
  160 this is record no 160  
  170 this is record no 170  
  180 this is record no 180  
>>> 190 this is record no 190  
  200 this is record no 200  
  210 this is record no 210  
  220 this is record no 220  
  230 this is record no 230  
  240 this is record no 240  
  250 this is record no 250  
  260 this is record no 260  
  270 this is record no 270  
  280 this is record no 280  
  290 this is record no 290  
  
requests H 170: 890 _____  
_____
```

Demandes "Ligne"

La grille affichée après l'exécution de la demande est la suivante :

```
b=MAIN m=LIB: DEM DAT > ln=2470                               ctl _____
  90 THIS IS RECORD NO      90
 100 this is record no     100
 110 this is record no     110
 120 this is record no     120
 130 this is record no     130
 140 this is record no     140
 150 THIS IS RECORD NO     150
 160 THIS IS RECORD NO     160
                                3 hidden line(s)
>>>> 200 this is record no 200
                                63 hidden line(s)
 900
 910 this is record no     910
 920 this is record no     920
 930 this is record no     930
 940 this is record no     940
 950 this is record no     950
 960 this is record no     960
 970 this is record no     970
 980 this is record no     980
      this is record no
requests _____
```

A noter que la ligne courante n'est jamais cachée.

4.22 DEMANDE HL

La demande HL (HighLight) permet d'afficher en vidéo inverse les lignes (non cachées) désignées par le paramètre adresse.

Paramètres de la demande :

paramètre adresse : ensemble de lignes
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Cette demande fait passer en mode "vidéo inverse".

Les grilles contenant les lignes spécifiées défilent jusqu'à ce que la dernière ligne indiquée ait été détectée ou une barre oblique introduite dans la zone de gestion (ctl) pour mettre fin au mode "vidéo inverse".

Une fois la demande exécutée et ce, dans tous les cas, le mode "vidéo inverse" est annulé et la ligne courante est la dernière ligne qui a été affichée en vidéo inverse.

Exemples :

```
hl 7:120=/Jean/
```

Les lignes de la plage de lignes 7 à 120 qui contiennent la chaîne de caractères "Jean" sont affichées en vidéo inverse.

```
hl CNTEXT
```

Les lignes qui ont CNTEXT comme identificateur de contexte sont affichées en vidéo inverse.

4.23 DEMANDE I

La demande I (Insert) permet d'insérer des lignes non formatées avant une ligne spécifiée.

Paramètres de la demande :

paramètre adresse : ligne unique
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Cette demande fait passer en mode saisie. (Pour plus de détails sur ce mode, voir le paragraphe 3.8).

Après l'exécution de la demande, la ligne courante est la dernière ligne introduite.

Remarque importante : Pour sortir du mode saisie, placer le curseur dans la zone de gestion (ctl) de la grille (en appuyant sur la touche ORIGINE - HOME), puis introduire un point-virgule (;).

Exemples :

i 640

Passage en mode saisie, l'insertion étant effectuée avant la ligne 640.

i

Passage en mode saisie, l'insertion étant effectuée avant la ligne courante.

i ^

Passage en mode saisie, l'insertion étant effectuée avant la première ligne du tampon.

4.24 DEMANDE IB

La demande IB (Insert Buffer) permet d'insérer le contenu d'un tampon auxiliaire et ce, un certain nombre de fois, avant chacune des lignes d'un ensemble de lignes spécifié.

Paramètres de la demande :

```

paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires :
  premier paramètre       : tampon
    valeur implicite      : aucune (ce paramètre est
                          obligatoire)
  second paramètre        : facteur de répétition
    valeur implicite      : 1

```

Le premier paramètre supplémentaire fournit le nom du tampon auxiliaire contenant les lignes à copier.

Le second paramètre supplémentaire est le facteur de répétition et doit être un entier positif.

Après l'exécution de la demande, la ligne courante est la dernière ligne insérée.

Exemples :

```
ib 20 DATA
```

Le contenu du tampon DATA est inséré avant la ligne 20.

```
ib 1:12 DATA
```

Le contenu du tampon DATA est inséré avant chacune des lignes 1 à 12.

```
ib 20:120 DATA 4
```

Quatre exemplaires du contenu du tampon DATA sont insérés avant chacune des lignes 20 à 120.

4.25 DEMANDE IF

La demande IF (Insert Formatted) permet d'insérer des lignes en mode formaté avant une ligne spécifiée.

Paramètres de la demande :

paramètre adresse : ligne unique
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Cette demande appelle l'application (fournie par le système et correspondant au langage de l'unité), pour introduire les nouvelles lignes en mode formaté.

Quand les lignes à insérer ont été introduites, placer une barre oblique (/) dans la zone EXIT pour sortir du mode saisie formatée.

Les caractères non graphiques ne sont pas acceptés avec cette demande. Si le texte à insérer en contient, il faut utiliser la demande I (Insertion).

Après exécution de la demande, la ligne courante est la dernière ligne introduite ou, si aucune ligne n'a été introduite, est inchangée.

Les formats standards des langages sont acceptés (voir le tableau du paragraphe 3.9).

Si une ligne plus longue que ne l'autorise le format standard correspondant est introduite, elle est tronquée.

Remarque importante :

Pour sortir du mode saisie formatée, placer le curseur dans la zone EXIT en appuyant sur la touche ORIGINE - HOME) et introduire une barre oblique (/).

Exemple :

```
if 640
```

Passage en mode saisie formatée, l'introduction étant effectuée avant la ligne 640.

4.26 DEMANDES IL1, IL2, IL3

Les demandes IL1 (InLib1), IL2 (InLib2) et IL3 (InLib3) sont utilisées pour affecter les bibliothèques d'entrée correspondantes.

Paramètres de la demande :

paramètre nom : nom de la bibliothèque
paramètres supplémentaires : aucun

Le paramètre "nom" spécifie la bibliothèque d'entrée à affecter. Si une bibliothèque a déjà été affectée, elle sera remplacée par la bibliothèque spécifiée.

Si aucun nom n'est fourni, la bibliothèque d'entrée correspondante est libérée.

Lorsque FSE est appelé à partir de certains processeurs (LIBMAINT, par exemple), cette demande ne peut pas être utilisée.

Cette demande n'a d'effet que pour FSE.

Exemples :

```
i11
```

La bibliothèque d'entrée courante de FSE est libérée.

```
i11 sd3.sl
```

La bibliothèque sd3.sl est affectée en tant que bibliothèque d'entrée de FSE.

```
i12 sd3.sl-trans
```

La bibliothèque sd3.sl-trans est affectée en tant que bibliothèque d'entrée de FSE.

4.27 DEMANDE IND

La demande IND (INDent) permet d'avoir des retraits dans les procédures GPL. L'exécution de cette demande s'arrête dès que la première erreur de syntaxe est détectée.

Paramètres de la demande :

```
paramètre adresse           : aucun
paramètres supplémentaires :
  premier paramètre        : langage
  valeur implicite         : langage associé
```

Le premier paramètre supplémentaire spécifie le langage de l'unité. La valeur implicite est le langage associé. Dans cette version, la fonction "retrait" est fournie uniquement pour les procédures GPL.

La valeur des paramètres LM, IN, CM et AT, qui peut être spécifiée dans la commande INDENT du processeur LIBMAINT, ne peut pas l'être dans cette demande pour laquelle les valeurs implicites de cette commande de LIBMAINT sont retenues (c'est-à-dire, LM=11, IN=5, CM=61 et AT=31).

Si aucune erreur n'est détectée pendant l'exécution de la demande, la ligne courante est inchangée. Dans le cas contraire, c'est la ligne qui contient la première erreur qui devient la ligne courante.

Exemple :

```
ind
```

Le contenu du tampon courant fera l'objet de retraits.

4.28 DEMANDE K

La demande K (copy) est utilisée pour copier un ensemble de lignes spécifié, après la ligne courante d'un tampon auxiliaire.

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires :
  premier paramètre       : tampon
  valeur implicite        : aucune (ce paramètre est
                           obligatoire)
```

Le premier paramètre supplémentaire est le nom du tampon auxiliaire dans lequel l'ensemble de lignes spécifié doit être copié.

Ces lignes sont ajoutées après la ligne courante du tampon auxiliaire.

Si ce tampon n'existe pas, il est créé.

Les numéros de lignes d'origine sont conservés.

La ligne courante (dans le tampon courant) est inchangée.

La dernière ligne copiée devient la nouvelle ligne courante dans le tampon auxiliaire.

Exemples :

```
k 17.2 DRAW_FIG
```

Une copie de la ligne 17.2 est ajoutée après la ligne courante dans le tampon auxiliaire DRAW_FIG.

```
k (1,8,14) EXEC_JOB
```

Une copie des lignes 1, 8 et 14 est ajoutée après la ligne courante dans le tampon auxiliaire EXEC_JOB.

4.29 DEMANDE KA

La demande KA (Copy Append) permet de copier un ensemble de lignes et de l'ajouter, un certain nombre de fois, après une ligne spécifiée.

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires :
  premier paramètre       : ligne unique
    valeur implicite     : la ligne courante
  second paramètre        : facteur de répétition
    valeur implicite     : 1
```

Le premier paramètre supplémentaire indique la ligne après laquelle les lignes copiées doivent être ajoutées. Une ligne unique doit être spécifiée.

L'identificateur de contexte et l'identificateur "." peuvent être utilisés.

Rappel : Le caractère '=' n'est valable que pour le paramètre adresse et ne peut donc pas être utilisé comme premier paramètre supplémentaire.

Le second paramètre supplémentaire est le facteur de répétition et doit être un entier positif.

Après exécution de la demande, la ligne courante est la dernière ligne ajoutée.

Exemples :

```
ka 3:18 300
```

Les lignes 3 à 18 sont copiées après la ligne 300.

```
ka 5:7 20 2
```

Deux copies des lignes 5 à 7 sont insérées après la ligne 20.

4.30 DEMANDE KI

La demande KI (Copy Insert) permet de copier un ensemble de lignes et de l'insérer, un certain nombre de fois, avant une ligne spécifiée.

Paramètres de la demande :

```

paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires :
  premier paramètre       : ligne unique
    valeur implicite      : la ligne courante
  second paramètre        : facteur de répétition
    valeur implicite      : 1

```

Le premier paramètre supplémentaire indique la ligne avant laquelle les lignes copiées doivent être insérer. Une ligne unique doit être spécifiée.

L'identificateur de contexte et l'identificateur "." peuvent être utilisés.

Rappel : Le caractère '=' n'est valable que pour le paramètre adresse et ne peut donc pas être utilisé comme premier paramètre supplémentaire.

Le second paramètre supplémentaire est le facteur de répétition et doit être un entier positif.

Après exécution de la demande, la ligne courante est la dernière ligne insérée.

Exemples :

```
ki 3:18 300
```

Les lignes 3 à 18 sont copiées avant la ligne 300.

```
ka 5:7 20 2
```

Deux copies des lignes 5 à 7 sont insérées avant la ligne 20.

4.31 DEMANDE KR

La demande KR (Copy Replace) est utilisée pour copier un ensemble de lignes, qui remplace le contenu d'un tampon auxiliaire spécifié.

Paramètres de la demande :

paramètre adresse	:	ensemble de lignes
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	
premier paramètre	:	tampon
valeur implicite	:	aucune (ce paramètre est obligatoire)

Le premier paramètre supplémentaire spécifie le nom du tampon dont le contenu doit être remplacé par les lignes copiées.

Si ce tampon n'existe pas, il est créé.

Les numéros de lignes d'origine sont conservés.

La ligne courante (dans le tampon courant) est inchangée.

La dernière ligne copiée devient la nouvelle ligne courante du tampon auxiliaire.

Exemple :

```
kr 1:3 DATA
```

Le contenu du tampon auxiliaire DATA est remplacé par une copie des lignes 1 à 3.

4.32 DEMANDE L

La demande L (Locate) permet de choisir comme ligne courante (localiser) la ligne spécifiée par le paramètre adresse et de visualiser le texte qui est juste avant et juste après cette nouvelle ligne courante.

Paramètres de la demande :

paramètre adresse : adresse unique
 valeur implicite : ligne courante
 paramètres supplémentaires : aucun

Pour un meilleur confort oculaire, les déplacements sur l'écran sont minimisés. Pour toute autre demande que la demande L, l'écran précédent est gardé si la ligne courante, après exécution de la demande, apparaît dans cet écran. Autrement, un nouvel écran centré autour de la ligne courante est constitué.

Note : Si la nouvelle ligne courante est la première ou la dernière ligne du tampon, la ligne apparaît soit en haut soit en bas (respectivement) de la zone texte.

Exemples :

l 380

La ligne 380 devient la nouvelle ligne courante.

l ^

La première ligne du tampon courant devient la nouvelle ligne courante.

l \$

La dernière ligne du tampon courant devient la nouvelle ligne courante.

l +40

La 40ème ligne après la ligne courante devient la nouvelle ligne courante.

l -20

La 20ème ligne avant la ligne courante devient la nouvelle ligne courante.

l /Exemple/

Une ligne contenant la chaîne de caractères "Exemple" devient la nouvelle ligne courante. (Recherche en aval, à partir de la ligne courante incluse).

l -/Exemple/

Demandes "Ligne"

Une ligne contenant la chaîne de caractères "Exemple" devient la nouvelle ligne courante. (Recherche en amont, à partir de la ligne courante incluse).

```
1 [#2/Exemple/
```

Une ligne contenant la deuxième occurrence de la chaîne de caractères "Exemple" devient la nouvelle ligne courante. (Recherche en aval, à partir de la ligne courante incluse).

```
1 -/^Exemple/
```

La première ligne contenant la chaîne de caractères "Exemple" en début de ligne devient la nouvelle ligne courante. (Recherche en amont, à partir de la ligne courante incluse).

```
1 [^/Exemple/
```

La première ligne NE contenant PAS la chaîne de caractères "Exemple" devient la nouvelle ligne courante. (Recherche en aval, à partir de la ligne courante incluse).

```
1 -[^/^Exemple/
```

La première ligne NE contenant PAS la chaîne de caractères "Exemple" en début de ligne devient la nouvelle ligne courante. (Recherche en amont, à partir de la ligne courante incluse).

```
1 -/Exemple$/
```

La première ligne contenant la chaîne de caractères "Exemple" en fin de ligne devient la nouvelle ligne courante. (Recherche en amont, à partir de la ligne courante incluse).

```
1 /Exemple/ [&/PROC/
```

La première ligne contenant la chaîne de caractères "Exemple" ET la chaîne de caractères "PROC" devient la nouvelle ligne courante. (Recherche en aval, à partir de la ligne courante incluse).

```
1 /Exemple/ [!/PROC/
```

La première ligne contenant la chaîne de caractères "Exemple" OU la chaîne de caractères "PROC" devient la nouvelle ligne courante. (Recherche en aval, à partir de la ligne courante incluse).

4.33 DEMANDE LE

La demande LE (Line Edit) permet de sortir du mode écran et de repasser en mode ligne.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

le

Passage en mode ligne.

4.34 DEMANDE LIB

La demande LIB (LIBrary) permet d'affecter la bibliothèque de sortie.

Paramètres de la demande :

paramètre nom : nom de la bibliothèque
paramètres supplémentaires : aucun

Le paramètre "nom" indique la bibliothèque de sortie qui doit être affectée. Si une bibliothèque de sortie est déjà affectée, elle est remplacée par la bibliothèque spécifiée.

Si aucun paramètre n'est fourni, la bibliothèque de sortie est libérée.

Si des unités de bibliothèque ont été modifiées, FSE demande si ces unités doivent ou non être sauvegardées dans la bibliothèque de sortie précédente.

- si la réponse est Y (Oui), il n'y aura pas de nouvelle affectation de bibliothèque.
- si la réponse est N (Non), la demande sera exécutée.

Cette demande n'est pas autorisée lorsque FSE est appelé à partir de certains processeurs (comme LIBMAINT, par exemple).

Cette demande n'a d'effet que pendant l'exécution de FSE.

Exemples :

```
lib
```

La bibliothèque de sortie courante de FSE est libérée.

```
lib sd3.sl
```

La bibliothèque sd3.sl est affectée en tant que bibliothèque de sortie de FSE.

4.35 DEMANDE LO

La demande LO (LOWer) permet de convertir en minuscules les lignes spécifiées par le paramètre adresse.

Paramètres de la demande :

```
paramètre adresse      : ensemble de lignes
valeur implicite      : la ligne courante
paramètres supplémentaires : aucun
```

Après l'exécution de la demande, la ligne courante est la dernière ligne modifiée.

Exemple d'utilisation de LO.

Avant l'exécution de LO, la grille affichée se présente comme suit :

```

b=MAIN m=LIB: DEM DAT > ln=2470                                ctl  ____
    90 THIS IS RECORD NO          90
    100 this is record no         100
    110 this is record no         110
    120 this is record no         120
    130 this is record no         130
    140 this is record no         140
    150 THIS IS RECORD NO         150
    160 THIS IS RECORD NO         160
    3 hidden line(s)
>>>> 200 this is record no         200
    63 hidden line(s)
    900 this is record no          900
    910 this is record no          910
    920 this is record no          920
    930 this is record no          930
    940 this is record no          940
    950 this is record no          950
    960 this is record no          960
    970 this is record no          970
    980 this is record no          980
    this is record no
requests _____
_____

```

Demandes "Ligne"

La grille affichée après l'exécution de la demande LO est la suivante :

```
b=MAIN m=LIB: DEM DAT > ln=2470                                ctl  _____

    100  THIS IS RECORD NO 100
    110  THIS IS RECORD NO 110
    120  THIS IS RECORD NO 120
    130  THIS IS RECORD NO 130
    140  THIS IS RECORD NO 140
    150  THIS IS RECORD NO 150
    160  THIS IS RECORD NO 160
    170  THIS IS RECORD NO 170
    180  THIS IS RECORD NO 180
    190  THIS IS RECORD NO 190
>>>> 200  THIS IS RECORD NO 200
    210  THIS IS RECORD NO 210
    220  THIS IS RECORD NO 220
    230  THIS IS RECORD NO 230
    240  THIS IS RECORD NO 240
    250  THIS IS RECORD NO 250
    260  THIS IS RECORD NO 260
    270  THIS IS RECORD NO 270
    280  THIS IS RECORD NO 280
    290  THIS IS RECORD NO 290

requests LO 150: 200  _____
_____
```

4.36 DEMANDE M

La demande M (Move) est utilisée pour transférer un ensemble de lignes spécifié après la ligne courante d'un tampon auxiliaire.

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires :
  premier paramètre       : tampon
  valeur implicite        : aucune (ce paramètre est
                           obligatoire)
```

Le premier paramètre supplémentaire spécifie le nom du tampon dans lequel les lignes seront transférées.

Ces lignes seront ajoutées après la ligne courante du tampon auxiliaire.

Si ce tampon n'existe pas, il est créé et les numéros de lignes d'origine sont conservés.

Après exécution de la demande, la ligne courante (dans le tampon courant) est la ligne qui suit la dernière ligne transférée. Si la ligne qui était la dernière a été transférée, c'est la nouvelle dernière ligne qui devient la ligne courante.

Au niveau du tampon auxiliaire, la dernière ligne transférée devient la ligne courante.

Exemples :

```
m 30:40 DATA
```

Les lignes 30 à 40 sont ajoutées à la fin du tampon auxiliaire DATA. (Ces lignes sont effacées là où elles figuraient précédemment)

```
m * EXJ
```

La totalité du contenu du tampon courant est ajoutée au tampon auxiliaire EXJ (le contenu du tampon courant étant effacé).

4.37 DEMANDE MA

La demande MA (Move Append) permet de transférer un ensemble de lignes, un certain nombre de fois, après une ligne spécifiée.

Paramètres de la demande :

paramètre adresse	:	ensemble de lignes
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	
premier paramètre	:	ligne unique
valeur implicite	:	la ligne courante
second paramètre	:	facteur de répétition
valeur implicite	:	1

Le premier paramètre supplémentaire indique la ligne après laquelle les lignes transférées doivent être ajoutées. Une ligne unique doit être spécifiée.

L'identificateur de contexte et l'identificateur "." peuvent être utilisés.

Rappel : Le caractère '=' n'est valable que pour le paramètre "adresse" et ne peut donc pas être utilisé comme premier paramètre supplémentaire.

Le second paramètre supplémentaire est le facteur de répétition et doit être un entier positif.

Après exécution de la demande, la ligne courante est la dernière ligne ajoutée.

Exemples :

```
ma 3:18 300
```

Les lignes 3 à 18 (incluses) sont transférées après la ligne 300. Cette opération entraîne l'effacement de ces lignes là où elles figuraient précédemment.

```
ma 5:7 20 2
```

Deux copies des lignes 5 à 7 (incluses) sont ajoutées après la ligne 20. Cette opération entraîne l'effacement de ces lignes là où elles figuraient précédemment.

4.38 DEMANDE MI

La demande MI (Move Insert) permet de transférer un ensemble de lignes, un certain nombre de fois, avant une ligne spécifiée.

Paramètres de la demande :

```

paramètre adresse           : ensemble de lignes
valeur implicite           : la ligne courante
paramètres supplémentaires :
premier paramètre          : ligne unique
valeur implicite           : la ligne courante
second paramètre           : facteur de répétition
valeur implicite           : 1

```

Le premier paramètre supplémentaire indique la ligne avant laquelle les lignes transférées doivent être insérées. Une ligne unique doit être spécifiée.

L'identificateur de contexte et l'identificateur "." peuvent être utilisés.

Rappel : Le caractère '=' n'est valable que pour le paramètre "adresse" et ne peut donc pas être utilisé comme premier paramètre supplémentaire.

Le second paramètre supplémentaire est le facteur de répétition et doit être un entier positif.

Après exécution de la demande, la ligne courante est la dernière ligne insérée.

Exemples :

```
mi 3:18 300
```

Les lignes 3 à 18 (incluses) sont transférées avant la ligne 300. Cette opération entraîne leur effacement là où elles figuraient précédemment.

```
ka 5:7 20 2
```

Les lignes 5 à 7 sont insérées deux fois avant la ligne 20. Cette opération entraîne leur effacement là où elles figuraient précédemment.

4.39 DEMANDE MR

La demande MR (Move Replace) est utilisée pour transférer un ensemble de lignes qui viennent remplacer le contenu d'un tampon auxiliaire.

Paramètres de la demande :

paramètre adresse	:	ensemble de lignes
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	
premier paramètre	:	tampon
valeur implicite	:	aucune (ce paramètre est obligatoire)

Le premier paramètre supplémentaire spécifie le nom du tampon dont le contenu sera remplacé par les lignes transférées.

Si ce tampon n'existe pas, il est créé.

Les numéros de lignes d'origine sont conservés.

Après exécution de la demande, la ligne qui suit la dernière ligne transférée devient la nouvelle ligne courante du tampon courant. Si la dernière ligne du tampon a été transférée, c'est la nouvelle dernière ligne qui devient la ligne courante.

Au niveau du tampon auxiliaire, c'est la dernière ligne transférée qui devient la nouvelle ligne courante.

Exemples :

```
mr 30:40 DATA
```

Le contenu du tampon auxiliaire DATA est remplacé par les lignes 30 à 40. (Cette opération provoque l'effacement des lignes là où elles figuraient précédemment).

```
mr * DRAWLINE
```

Le contenu du tampon auxiliaire DRAWLINE est remplacé par le contenu du tampon courant. Le transfert de ces lignes entraîne leur effacement dans le tampon courant.

4.40 DEMANDE NAME

La demande NAME permet de définir ou redéfinir les paramètres implicites des demandes de sauvegarde (SV et SVQ).

Paramètres de la demande :

paramètre adresse	:	aucun
paramètres supplémentaires	:	
premier paramètre	:	unité
valeur implicite	:	nom implicite
second paramètre	:	langage, longueur d'article
valeur implicite	:	langage implicite, pas d'articles de longueur fixe
troisième paramètre	:	remplacement ou non
valeur implicite	:	statut REPLACE implicite

Si aucun paramètre n'est spécifié, le nom d'unité implicite courant cesse d'être affecté.

Le premier paramètre supplémentaire fournit le (nouveau) nom associé au tampon courant.

Si le tampon courant a un nom implicite qui lui est associé, c'est celui qui est pris par défaut. Dans le cas contraire, ce paramètre est absolument obligatoire. (Voir au chapitre 1 les règles concernant le nom des unités).

Le second paramètre supplémentaire indique le langage de l'unité et la longueur de l'article, de la manière suivante :

<langage>

ou

([<langage>] , <longueur d'article>)

Remarque : Le nombre d'espaces n'a aucune signification particulière.

<langage> a pour valeur par défaut le langage implicite de l'unité (s'il y en a un), sinon, c'est DAT.

<longueur d'article> s'applique à des articles de longueur fixe et doit être un nombre de 1 à 255.

Les articles ayant une longueur supérieure à 255 caractères sont tronqués au moment de la sauvegarde.

Demandes "Ligne"

Si le paramètre <longueur d'article> n'est pas spécifié, la valeur implicite est qu'il ne s'agit pas d'articles de longueur fixe.

Ainsi,

```
GPL ( ,100) (COB, 80)
```

sont des seconds paramètres supplémentaires valables.

Le troisième paramètre supplémentaire permet de spécifier ou de modifier le statut REPLACE (remplacement ou non), sous la forme 0 ou 1.

Si le nom de l'unité n'est pas changé, la valeur par défaut est le statut REPLACE implicite, sinon la valeur par défaut est zéro (pas de remplacement).

Après exécution de la demande, la ligne courante reste inchangée.

Exemples :

```
nom
```

Supposons (pour cet exemple et les exemples suivants) que le nom implicite associé au tampon courant soit "nom_précédent". Une fois la demande exécutée, ce nom "nom_précédent" cesse d'être affecté au tampon courant.

```
name nom_nouveau
```

"nom_précédent" est changé en "nom_nouveau". Le langage associé est inchangé. Le statut REPLACE a la valeur 0 (pas de remplacement).

```
name nom_nouveau dat
```

"nom_précédent" est changé en "nom_nouveau". Le langage associé est DAT. Le statut REPLACE a la valeur 0 (pas de remplacement).

```
name ,(dat,80)
```

Le nom précédent est inchangé. Le langage est DAT. Le statut REPLACE est également inchangé. La longueur d'article utilisée au moment de la sauvegarde est 80.

```
name nom_nouveau (dat,80) 1
```

Le nom précédent est changé en "nom_nouveau". Le langage est DAT. Le statut REPLACE est 1 (remplacement). La longueur d'article utilisée au moment de la sauvegarde est 80.

4.41 DEMANDE NOV

La demande NOV (NOVice) permet de sortir du mode "expert" et de passer en mode "novice".

Ce mode est le mode implicite pour les utilisateurs débutants, c'est-à-dire pour les utilisateurs dont la variable GCL #NOVICE a la valeur 1.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaire : aucun

La ligne courante est inchangée.

Exemple :

nov

Passage en mode "novice".

Remarque : Pour sortir du mode "novice", introduire "exp".

4.42 DEMANDE OW

La demande OW (Open Window) permet d'ouvrir une deuxième fenêtre pour visualiser le contenu d'un tampon spécifié.

Seules les demandes de visualisation (L,>,<,R, par exemple) sont acceptées pour cette fenêtre dite de visualisation. Il faut donc charger le tampon avant d'ouvrir la fenêtre de visualisation ou introduire des demandes de lecture, copie ou transfert dans la zone "request" de la fenêtre active.

Paramètres de la demande :

paramètre adresse	:	aucun
paramètres supplémentaires	:	
premier paramètre	:	tampon
valeur implicite	:	tampon courant

Le premier paramètre supplémentaire est le nom du tampon pour lequel une fenêtre de visualisation doit être ouverte.

Voir au chapitre 3 le paragraphe concernant les noms de tampon.

Après exécution de la demande, la ligne courante est inchangée.

Exemples :

```
ow DATA
```

Une deuxième fenêtre est ouverte pour le tampon DATA.

```
ow main
```

Une deuxième fenêtre est ouverte pour le tampon en cours de modification (à supposer que, pour cet exemple, cette demande a été émise à partir du tampon principal (main)).

4.43 DEMANDE PF

La demande PF (Print Formatted) permet d'afficher, une par une, les lignes de l'ensemble de lignes spécifié, en utilisant l'application (fournie par le système) correspondant au langage utilisé.

Paramètres de la demande :

paramètre adresse : ensemble de lignes
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est la dernière ligne de l'ensemble de lignes spécifié.

Si l'exécution de la demande est interrompue (en introduisant une barre oblique (/) dans la zone "ctl"), la ligne courante est la ligne qui est affichée en bas de l'écran.

4.44 DEMANDE QC

La demande QC (Quit Context) permet de sortir du mode mise à jour de contexte (qui a été instauré par la demande UC) et de repasser en mode mise à jour normal.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Exemple :

qc

Passage en mode mise à jour normal.

4.45 DEMANDES QUIT, Q , /

La demande QUIT (ou Q et /) permet de sortir de la session MODIFY et de revenir au niveau commande (caractères de guidage F:)

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Si au moins un tampon associé à une unité a été modifié depuis la dernière demande de sauvegarde , la question suivante est affichée (la réponse devant être introduite dans la zone "ctl").

"DO YOU WANT TO SAVE BEFORE LEAVING ? ANSWER Y(ES) OR N(O).

(Voulez-vous faire une sauvegarde avant de sortir ? Répondre Y(ES) ou N(O))

Si la réponse est Y, la demande QUIT est ignorée et il est possible de sauvegarder n'importe quel tampon.

Si la réponse est N, la sortie de la session MODIFY se fait sans sauvegarde du ou des tampons modifié(s), les caractères de guidage F: s'affichant pour solliciter l'introduction d'une commande.

Exemple :

q

Sortie de la session MODIFY et passage au niveau commande FSE.

4.46 DEMANDE R

La demande R (Repeat) permet la répétition de la dernière demande "ligne" exécutée.

Les points suivants doivent être notés :

- Les demandes 'R' et '=' ne sont pas considérées comme la dernière demande exécutée.
- Les macro-demandes sont considérées comme des demandes exécutées. Ainsi, si la dernière demande était le nom d'un tampon d'exécution x, la demande R exécute à nouveau x.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

r

La dernière demande "ligne" exécutée est répétée.

4.47 DEMANDE RA

La demande RA (Read Append) permet de lire la totalité ou une partie seulement d'une unité d'une bibliothèque d'entrée et d'ajouter les lignes lues, après une ligne spécifiée.

Paramètres de la demande :

paramètre adresse	:	ligne unique
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	
premier paramètre	:	unité
valeur implicite	:	aucune (ce paramètre est obligatoire)
deuxième paramètre	:	bibliothèque
valeur implicite	:	LIB
troisième paramètre	:	de : à
valeur implicite	:	la totalité de l'unité

Le premier paramètre supplémentaire spécifie le nom de l'unité à lire.

Le second paramètre supplémentaire indique la bibliothèque dans laquelle l'unité doit être recherchée. Les identificateurs de bibliothèque sont LIB, IL1, IL2, IL3.

Le troisième paramètre supplémentaire permet une lecture partielle. La lecture commence quand il y a correspondance entre le premier numéro de ligne spécifié et le numéro de ligne détecté. Elle s'arrête lorsque la ligne dont le numéro correspond au dernier numéro ligne spécifié a été ajoutée.

A noter la syntaxe particulière de ce paramètre :

<premier numéro de ligne SSF> : <dernier numéro de ligne SSF>

Si les lignes lues sont placées dans un tampon qui est vide au départ, la première ligne devient la nouvelle ligne courante. Dans le cas contraire, c'est la dernière ligne lue qui devient la nouvelle ligne courante.

Si la lecture s'effectue dans un tampon sans nom d'unité associé, l'unité lue reçoit le nom de ce tampon. Si, dans ce cas :

- le tampon était vide avant la lecture, et
- il s'agit d'une lecture complète (et non pas partielle), et
- la lecture a été faite à partir de la bibliothèque LIB,

le statut REPLACE est forcé à 1 (remplacement), ce qui constitue la valeur implicite pour les sauvegardes ultérieures.

Dans le cas où l'une de ces conditions n'est pas remplie, le statut REPLACE est forcé à 0 (pas de remplacement).

Demandes "Ligne"

Exemples :

```
ra 19820 jh_ex
```

La totalité de l'unité jh_ex est lue et placée dans le tampon, après la ligne 19820.

```
ra $ new_doc
```

La totalité de l'unité existante new_doc est lue et placée après la dernière ligne du tampon courant.

```
ra $ new_doc, , 5:10 ;
```

Dans cet exemple, l'unité new_doc est lue à partir de la ligne SSF qui porte le numéro 5 jusqu'à la ligne SSF qui a le numéro 10. Si l'unité contient plus d'une séquence de lignes numérotées de 5 à 10, seule la première de ces séquences est lue. Le texte lu est copié à la fin du tampon courant.

```
ra $ new_doc ill 5:10 ;
```

Dans cet exemple, l'unité new_doc est lue à partir de la ligne SSF qui porte le numéro 5 jusqu'à la ligne SSF qui a le numéro 10. L'unité lue se trouve dans la bibliothèque d'entrée affectée à il1.

4.48 DEMANDE RB

La demande RB (RenumBer) permet de renuméroter une plage de lignes spécifiée.

Paramètres de la demande :

```
paramètre adresse           : plage de lignes
  valeur implicite         : la totalité du tampon.
paramètres supplémentaires :
  premier paramètre       : numéro de ligne initial
  valeur implicite        : 1
  deuxième paramètre      : incrément
  valeur implicite        : 1
```

Le premier paramètre supplémentaire spécifie le numéro à attribuer à la première ligne renumérotée.

Le second paramètre supplémentaire fournit l'incrément (pas de progression) à utiliser pour le calcul des numéros de ligne suivants.

Pour plus de détails en ce qui concerne la numérotation des lignes, voir le paragraphe 3.6.

Pour améliorer les performances lors de la définition d'une nouvelle ligne courante et de la constitution d'un nouvel écran, il est recommandé de renuméroter la totalité du tampon aussi souvent que possible.

Exemples :

```
rb 2860:3130 230 10
```

Re-numéroter les lignes 2860 à 3130, de 10 en 10, en commençant à 230.

```
rb
```

Re-numéroter toutes les lignes du tampon, en commençant à 1 et avec un incrément de 1 également.

4.49 DEMANDE REM

La demande REM (REMove context) est utilisée pour supprimer un ensemble de lignes spécifié d'un contexte précédemment affecté.

Paramètres de la demande :

paramètre adresse	:	ensemble de lignes
valeur implicite	:	la ligne courante
paramètres supplémentaires :		
premier paramètre	:	contexte
valeur implicite	:	aucune (ce paramètre est obligatoire)

Le premier paramètre supplémentaire est l'identificateur du contexte dont un ensemble de lignes doit être supprimé.

Voir au chapitre 3 le paragraphe concernant l'identificateur de contexte.

L'identificateur doit déjà exister (c'est-à-dire être affecté).

Le "type" donné à un identificateur de contexte par FSE après une demande REM est "ensemble de lignes", sauf si l'identificateur correspond à un contexte ne contenant qu'une seule ligne ou vide. Lorsque le contexte ne contient qu'une seule ligne, le type est "ligne unique". Lorsqu'il est vide, l'identificateur cesse d'être affecté.

Après exécution de la demande, la ligne courante est inchangée.

Exemples :

```
rem 410 ALL_PROC
```

Suppression de la ligne 410 dans l'ensemble de lignes spécifié par l'identificateur de contexte ALL_PROC.

```
rem . ALL_PROC
```

Suppression de la ligne courante dans l'ensemble de lignes spécifié par l'identificateur de contexte ALL_PROC.

```
rem 6:10 ALL_PROC
```

Suppression des lignes 6 à 10 dans l'ensemble de lignes spécifié par l'identificateur de contexte ALL_PROC.

```
rem OLD_PROC ALL_ PROC
```

Suppression de l'ensemble de lignes spécifié par l'identificateur de contexte OLD_PROC dans l'ensemble de lignes désigné par l'identificateur de contexte ALL_PROC.

4.50 DEMANDE RI

La demande RI (Read Insert) permet de lire une unité d'une bibliothèque d'entrée et d'insérer les lignes lues avant une ligne spécifiée.

Une lecture partielle est possible, en indiquant la première et la dernière ligne à lire.

Paramètres de la demande :

paramètre adresse	:	ligne unique
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	
premier paramètre	:	unité de bibliothèque
valeur implicite	:	aucune (ce paramètre est obligatoire)
deuxième paramètre	:	bibliothèque
valeur implicite	:	LIB
troisième paramètre	:	de : à
valeur implicite	:	la totalité de l'unité

Le premier paramètre supplémentaire spécifie le nom de l'unité à lire.

Le second paramètre supplémentaire précise la bibliothèque dans laquelle l'unité doit être recherchée. Les identificateurs de bibliothèque sont LIB, IL1, IL2, IL3.

Le troisième paramètre supplémentaire permet une lecture partielle. La lecture commence quand il y a correspondance entre le premier numéro de ligne spécifié et le numéro de ligne détecté. Elle s'arrête lorsque la ligne dont le numéro correspond au dernier numéro ligne spécifié a été insérée.

A noter la syntaxe particulière de ce paramètre :

<premier numéro de ligne SSF> : <dernier numéro de ligne SSF>

Si les lignes lues sont placées dans un tampon qui est vide au départ, la première ligne devient la nouvelle ligne courante. Dans le cas contraire, c'est la dernière ligne lue qui devient la nouvelle ligne courante.

Si la lecture s'effectue dans un tampon sans nom d'unité associé, l'unité lue reçoit le nom de ce tampon. Si, dans ce cas :

- le tampon était vide avant la lecture, et
- il s'agit d'une lecture complète (et non pas partielle), et
- la lecture a été faite à partir de la bibliothèque LIB,

le statut REPLACE est forcé à 1 (remplacement), ce qui constitue la valeur implicite pour les sauvegardes ultérieures.

Demandes "Ligne"

Dans le cas où l'une de ces conditions n'est pas remplie, le statut REPLACE est forcé à 0 (pas de remplacement).

Exemples :

```
ri 19820 jh_ex
```

La totalité de l'unité jh_ex est lue et placée dans le tampon avant la ligne 19820.

```
ri ^ new_doc il1 5:10 ;
```

L'unité new_doc est lue à partir de la ligne SSF qui porte le numéro 5 jusqu'à la ligne SSF qui a le numéro 10. Les lignes sont lues dans la bibliothèque d'entrée affectée à il1 et placées au début du tampon courant.

```
d 5:10;ri . new_doc, , 5:10 ;
```

Dans cet exemple, une copie des lignes d'origine doit remplacer toutes les modifications apportées aux lignes 5 à 10. Ces lignes 5 à 10 sont donc supprimées, après quoi l'unité new_doc est lue de la ligne SSF portant le numéro 5 jusqu'à la ligne SSF ayant le numéro 10. A noter que si l'unité contient plus d'une séquence de lignes numérotées de 5 à 10, seule la première de ces séquences est lue. Le texte lu est placé exactement là où les lignes ont été supprimées.

4.51 DEMANDE S

La demande S (Substitute) permet de modifier le contenu d'un ensemble de lignes spécifié, en remplaçant les chaînes de caractères qui correspondent à une chaîne régulière donnée par une autre chaîne de caractères.

Paramètres de la demande :

paramètre adresse	:	ensemble de lignes
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	
premier paramètre	:	ancienne chaîne de caractères
valeur implicite	:	aucune (ce paramètre est obligatoire)
deuxième paramètre	:	nouvelle chaîne de caractères
valeur implicite	:	aucune (ce paramètre est obligatoire)

L'ancienne chaîne de caractères est une chaîne régulière, telle qu'elle a été définie au chapitre 3.

S'il y a plus d'une chaîne correspondant à la chaîne régulière, c'est la chaîne la plus à gauche qui est retenue.

S'il y a eu substitution d'un ou de plusieurs caractères dans une ligne, la substitution est tentée à nouveau pour le reste de la ligne.

Il est possible d'utiliser le caractère "=" à la place du paramètre "ancienne chaîne" dans la demande S.

La nouvelle chaîne est délimitée par deux barres obliques (/) ou par deux apostrophes (').

Le caractère & a une signification spéciale; il est remplacé au moment de la substitution par la chaîne de caractères qui correspond au premier paramètre.

Quand la séquence d'échappement [X est suivie du code hexadécimal d'un caractère, elle équivaut à ce caractère et peut ainsi être utilisée pour représenter des caractères non graphiques.

Il est possible d'utiliser le caractère "=" à la place du dernier paramètre "nouvelle chaîne" dans la demande S.

Dans la nouvelle chaîne de caractères, la signification spéciale de "/", "&", "' '", "[C", "[X", "[V", "=" peut être supprimée au moyen de la séquence d'échappement [C.

Demandes "Ligne"

Ainsi,

[C&	désigne le caractère &.
[C[V (#2) ou [[CV(#2)	désigne la chaîne de caractères [V(#2).

Après exécution de cette demande, la ligne courante est la dernière ligne modifiée

Si l'ancienne chaîne de caractères (ou la nouvelle) ne contient aucun séparateur (virgule, guillemet ou point-virgule) ou aucun délimiteur (en début de chaîne) et si celle-ci n'est pas égale au caractère spécial "=", il est possible d'omettre les délimiteurs (apostrophe ou barre oblique).

Exemples :

```
s 740 3 2
```

Dans la ligne 740, 3 est remplacé par 2.

```
S . Jean Peter
```

Dans la ligne courante, la chaîne de caractères "Jean" est remplacée par la chaîne de caractères "Peter".

```
S . /Jean Brun/ /John Black/
```

Dans la ligne courante, la chaîne de caractères "Jean Brun" est remplacée par la chaîne de caractères "John Black".

```
S . /^/ /John Black/
```

La chaîne de caractères "John Black" est insérée au début de la ligne courante.

```
S . /^.. / /John Black/
```

Dans la ligne courante, les deux premiers caractères sont remplacés par la chaîne "John Black"

```
S . /$/ /John Black/
```

La chaîne de caractères "John Black" est ajoutée à la fin de la ligne courante.

```
S . /....$/ /John Black/
```

Les quatre derniers caractères de la ligne courante sont remplacés par la chaîne de caractères "John Black".

```
S . /Jean/ //
```

Dans la ligne courante, toutes les occurrences de la chaîne de caractères "Jean" sont supprimées.

```
S * /autre tampon/ /cet &/
```

Dans tout le tampon, la chaîne "autre tampon" devient "cet autre tampon".

```
s 20 /A..B/ /ABLE/
```

Chaque occurrence, dans la ligne 20, de A suivi de deux caractères et de B est remplacée par la chaîne de caractères ABLE.

```
s 0.31 / */ / /
```

Chaque occurrence, dans la ligne 0.31, de deux ou plusieurs espaces est remplacée par un seul espace.

```
s . /. / / /
```

Tous les caractères de la ligne courante sont remplacés par des espaces.

```
s * 'LMARGE=15' 'LNOMBRE'
```

Remplacement de LMARGE=15 par LNOMBRE dans tout le tampon.

```
s 740:790 '/' '#'
```

Remplacement de / par #, dans les lignes 740 à 790.

```
s 740:790 /#/ '/'
```

Remplacement de # par ' dans les lignes 740 à 790.

```
s (1,4,13 8:11) Jim Pat
```

Dans les lignes 1, 4, 13 et 8 à 11, la chaîne de caractères "Jim" est remplacée par la chaîne de caractères "Pat".

```
s 14.5:14.9=/Exemple/Jim Pat
```

Dans toutes les lignes de 14.5 à 14.9 qui contiennent la chaîne de caractères "Exemple", la chaîne de caractères "Jim" est remplacée par la chaîne de caractères "Pat".

```
s *=[^/Exemple/ Jim Pat
```

Dans toutes les lignes du tampon qui ne contiennent pas la chaîne de caractères "Exemple", la chaîne de caractères "Jim" est remplacée par la chaîne de caractères "Pat".

```
s */[^Exemple/ Jim Pat
```

Dans toutes les lignes du tampon qui commencent par la chaîne "Exemple", la chaîne de caractères "Jim" est remplacée par la chaîne de caractères "Pat".

```
s 14.5:14.9=/Exemple/ [&/ quatre/ sept huit
```

Demandes "Ligne"

Dans toutes les lignes de 14.5 à 14.9 qui contiennent la chaîne de caractères "Exemple" et la chaîne de caractères "quatre", la chaîne "sept" est remplacée par la chaîne "huit".

```
s 14.5:14.9=/Exemple/[!/quatre/ sept huit
```

Dans toutes les lignes de 14.5 à 14.9 qui contiennent la chaîne de caractères "Exemple" ou la chaîne de caractères "quatre", la chaîne "sept" est remplacée par la chaîne "huit".

```
s PROC pair impair
```

Dans toutes les lignes désignées par l'identificateur de contexte PROC, la chaîne de caractères "pair" est remplacée par la chaîne de caractères "impair".

4.52 DEMANDE SB

La demande SB (Switch Buffer) permet de changer de tampon.

Paramètres de la demande :

paramètre adresse	:	aucun
paramètres supplémentaires	:	
premier paramètre	:	tampon
valeur implicite	:	aucune (ce paramètre est obligatoire)

Le premier paramètre supplémentaire spécifie le nom du nouveau tampon courant. En ce qui concerne les noms de tampon, voir paragraphe 3.11.2).

Quand la fenêtre de visualisation est ouverte et que le tampon sur lequel l'utilisateur veut basculer est associé à la fenêtre de visualisation, la fenêtre active devient la fenêtre de visualisation, et la fenêtre de visualisation la fenêtre active.

Les lignes courantes dans l'ancien et le nouveau tampon courant sont inchangées.

Exemples :

```
sb BUF1
```

Passage à un nouveau tampon dont le nom est BUF1.

```
sb main
```

Retour au tampon de travail initial.

4.53 DEMANDE SE

La demande SE (Screen Edit) permet de passer en mode écran (Voir au chapitre 5 le mode écran).

Pour sortir du mode écran, utiliser la demande LE.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

se

Passage en mode écran.

4.54 DEMANDE SEND

La demande SEND permet d'envoyer au système des commandes ou directives GCL.

Paramètres de la demande :

paramètre adresse	:	aucun
paramètres supplémentaires	:	
premier paramètre	:	cmde_gcl
valeur implicite	:	aucune (ce paramètre est obligatoire)

Le premier paramètre supplémentaire est le texte de la commande GCL, sous la forme d'une chaîne de caractères délimitée par deux barres obliques (/) ou par deux apostrophes (').

Si le texte en question ne contient ni séparateurs (virgule ou point-virgule), ni (en début de chaîne, et aucun délimiteur, il est possible d'omettre les délimiteurs pour ce paramètre.

Exemples :

```
SEND 'ejr my_job lib=my_lib holdout';
SEND 'dld load; dj';
SEND mail;
```

En ce qui concerne le dernier exemple, la commande GCL 'mail' doit être utilisée lorsque le message MAIL WAITING (Courrier en attente) est affiché sur la dernière ligne de l'écran.

4.55 DEMANDE SP

La demande SP (SPlit) permet de couper, en deux lignes ou plus, chaque ligne d'un ensemble de lignes, dès lors qu'un profil donné a été détecté. La coupure se fait juste avant le premier caractère du profil.

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires :
  premier paramètre       : profil (chaîne)
  valeur implicite        : aucune (ce paramètre est
                           obligatoire)
```

Le premier paramètre supplémentaire spécifie le profil (chaîne) à rechercher.

Ce profil est une chaîne régulière, telle qu'elle est définie au chapitre 3.

Chaque ligne est coupée juste avant le premier caractère de la chaîne. La comparaison se poursuit sur le reste de la ligne afin d'effectuer éventuellement d'autres coupures.

A noter que les chaînes telles que `/x*/` (x étant n'importe quel caractère), `/.*/`, `/^/` et `/$/` ne peuvent pas être utilisées comme indicateurs de coupure de ligne.

Si la chaîne ne contient ni séparateur (virgule ou point-virgule) ni (en début de chaîne et aucun délimiteur, il est possible d'omettre l'apostrophe ou la barre oblique pour délimiter cette chaîne.

Après exécution de la demande, la ligne courante est la dernière ligne de l'ensemble de lignes, sauf si la dernière ligne a été coupée, auquel cas la ligne courante correspond à la dernière partie seulement de cette ligne.

Exemple :

```
sp . /AB/ (la ligne courante étant 'AAAABAAAAABC')
```

La ligne courante est coupée à deux niveaux, ce qui donne les trois lignes suivantes :

```
'AAA '
'ABAAAA '
'ABC '
```

4.56 DEMANDE SQ

La demande SQ (Substitute Query) permet de modifier le contenu d'un ensemble de lignes, en remplaçant toutes les chaînes correspondant à un chaîne régulière donnée par une autre chaîne spécifiée.

Pour chaque ligne de l'ensemble contenant la chaîne recherchée, FSE affiche cette ligne et demande ce qu'il doit faire au moyen du message suivant :

'TRANSMIT TO VALIDATE, "/" TO GIVE UP, "S" TO SKIP, "L" TO LEAVE QUERY MODE'.

(Appuyer sur TRANS pour valider, / pour abandonner, S pour sauter cette ligne et L pour sortir du mode demande de confirmation).

Si l'utilisateur appuie seulement sur la touche de transmission (TRANS) (la zone ctl restant vide), la ligne courante est modifiée et l'exécution se poursuit normalement, la ligne suivante susceptible d'être modifiée s'affichant à l'écran ainsi que le message décrit précédemment.

Si la zone ctl est forcée à :

- "/" :	La ligne n'est pas modifiée et la demande est rejetée.
- "S" :	La ligne n'est pas modifiée mais l'exécution continue.
- "L" :	La ligne est modifiée et les modifications suivantes sont effectuées sans demande de confirmation.

Paramètres de la demande :

paramètre adresse	: ensemble de lignes
valeur implicite	: la ligne courante
paramètres supplémentaires :	
premier paramètre	: ancienne chaîne
valeur implicite	: aucune (ce paramètre est obligatoire)
deuxième paramètre	: nouvelle chaîne
valeur implicite	: aucune (ce paramètre est obligatoire)

Pour plus de détails sur les paramètres, se reporter à la demande S.

Exemple :

```
sq * Jean Peter
```

Lors de la détection de la première occurrence de la chaîne "Jean", FSE émet le message décrit ci-dessus. Si l'utilisateur appuie sur la touche de transmission (TRANS), la chaîne de caractères "Jean" est remplacée par la chaîne de caractères "Peter". S'il frappe S, la chaîne de caractères reste inchangée. Dans les deux cas, la prochaine occurrence de la chaîne de caractères "Jean" est recherchée et, si elle est détectée, le message ci-dessus s'affiche à nouveau.

4.57 DEMANDE STP

La demande STP (STeP) est utilisée pour passer en mode pas à pas.

Lorsque ce mode est en vigueur, une grille de guidage s'affiche pour chaque demande "ligne" avant qu'elle soit exécutée. Ce mode est tout particulièrement utile pour la mise au point des macro-demandes.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Il est possible de modifier les paramètres, d'introduire le caractère point-virgule (;) dans la zone de gestion (ctl) pour modifier la ligne demande ou le caractère barre oblique (/) pour effacer la ligne demande et sortir.

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

```
stp
```

Passage en mode pas à pas.

4.58 DEMANDE SV

La demande SV (SaVe) permet de sauvegarder dans la bibliothèque de sortie tout ou partie du contenu du tampon courant.

Paramètres de la demande :

paramètre adresse	: plage de lignes
valeur implicite	: * (la totalité du tampon).
paramètres supplémentaires	:
premier paramètre	: nom unité
valeur implicite	: nom implicite
deuxième paramètre	: langage, longueur d'article
valeur implicite	: langage implicite, pas d'articles de longueur fixe
troisième paramètre	: remplacement ou non
valeur implicite	: statut REPLACE implicite

Le premier paramètre supplémentaire spécifie le nom de l'unité dans laquelle le contenu du tampon doit être sauvegardé.

Si un nom implicite est associé au tampon, ce nom constitue la valeur implicite; sinon, le premier paramètre est obligatoire. (Voir les règles concernant les noms d'unité au chapitre 1).

Le deuxième paramètre indique le langage et la longueur d'article comme suit :

<langage> ou ([<langage>] , <longueur d'article>)

A noter que le nombre d'espaces n'a aucune signification particulière.

<langage> : La valeur par défaut de ce paramètre est le langage implicite; sinon, c'est DAT.

<longueur d'article> : Il s'agit de celle d'articles de longueur fixe (de 1 à 255). Les articles dont la longueur est supérieure à cette valeur sont tronqués.

Si la sauvegarde est effectuée en utilisant le nom implicite de l'unité, la longueur par défaut pour ce paramètre est la longueur d'article implicite correspondante. Dans le cas contraire, la valeur implicite est qu'il ne s'agit pas d'articles de longueur fixe.

Ainsi,

GPL (, 100) (COB, 80)

sont des deuxièmes paramètres valables.

Demandes "Ligne"

Le troisième paramètre supplémentaire est utilisé pour préciser ou modifier le statut REPLACE qui peut être 0 ou 1. Quand la valeur fournie est 1 (remplacement) et que l'unité existe déjà dans la bibliothèque, cette unité est recouverte. Quand la valeur fournie est 0 (pas de remplacement), l'unité ne doit pas exister dans la bibliothèque (sinon, la sauvegarde est impossible et l'utilisateur est invité à choisir la valeur 1).

Si la sauvegarde est effectuée en utilisant le nom implicite de l'unité, la valeur par défaut de ce paramètre est le statut REPLACE implicite correspondant; sinon, la valeur implicite est 0.

Remarque : Les paramètres implicites (nom d'unité, langage, et longueur d'article) peuvent être définis ou redéfinis au moyen des demandes RA, RI et NAME. Ils ne sont pas modifiés par les demandes SV ou SVQ.

Après exécution de la demande, la ligne courante est inchangée.

Exemples :

```
sv
```

La totalité du tampon est rangée dans l'unité ayant comme nom implicite celui qui est associé au tampon.

```
sv 640:730 UNITNOUV
```

Les lignes 640 à 730 sont rangées dans UNITNOUV.

```
sv * UNITANC , ,1
```

Le contenu de l'unité existante UNITANC est recouvert par la totalité du contenu du tampon.

```
sv * UNITANC (DAT,60) 1
```

Le contenu de l'unité existante UNITANC est recouvert par la totalité du contenu du tampon. Le langage est DAT et la longueur de l'article 60.

4.59 DEMANDE SVQ

La demande SVQ (SaVe and Quit) permet de sauvegarder dans la bibliothèque de sortie tout ou partie du tampon courant et de sortir de l'éditeur.

Paramètres de la demande :

paramètre adresse	:	plage de lignes
valeur implicite	:	* (la totalité du tampon)
paramètres supplémentaires	:	
premier paramètre	:	unité
valeur implicite	:	nom implicite
deuxième paramètre	:	langage, longueur
valeur implicite	:	d'article
troisième paramètre	:	langage implicite, pas
valeur implicite	:	d'articles de longueur
quatrième paramètre	:	fixe
cinquième paramètre	:	remplacement ou non
sixième paramètre	:	statut REPLACE implicite

Pour plus de détails sur les paramètres, se reporter à la demande SV. La seule différence avec la demande SV est qu'en cas de sortie de la session MODIFY après sauvegarde, les caractères de guidage F: sont visualisés.

Si au moins un des tampons associés à une unité a été modifié depuis la dernière demande de sauvegarde, la question suivante (dont la réponse doit être introduite dans la zone "ctl") est visualisée :

"DO YOU WANT TO SAVE BEFORE LEAVING ? ANSWER Y(ES) OR N(O)".

(Faut-il faire une sauvegarde avant de sortir ? Répondre Y(ES) ou N(O)).

Si la réponse est Y, il est possible de sauvegarder n'importe quel tampon. La ligne courante est inchangée.

Si la réponse est N, aucune sauvegarde du ou des tampon(s) modifié(s) ne précède la sortie de la session MODIFY.

Exemple :

```
svq
```

La totalité du tampon courant est sauvegardée dans l'unité dont le nom est implicite et il y a passage au niveau commande FSE.

4.60 DEMANDE TOP

La demande TOP permet de placer la fenêtre active en haut de l'écran.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

top

La fenêtre active est placée en haut de l'écran.

4.61 DEMANDE UC

La demande UC (Update Context) permet de passer en mode mise à jour de contexte, le contexte mis à jour pouvant être un ensemble de lignes, une plage de lignes ou une ligne unique. Tout se passe comme s'il s'agissait d'un tampon. (Pour plus de détails, voir au chapitre 3 le paragraphe concernant la mise à jour de contexte).

Pour sortir de ce mode, utiliser la demande QC.

Paramètres de la demande :

paramètre adresse	:	ensemble de lignes
valeur implicite	:	la ligne courante
paramètres supplémentaires	:	aucun

Après exécution de la demande, la première ligne du contexte devient la ligne courante.

Exemples :

```
uc *=/TITRE/
```

Passage en mode mise à jour de contexte. Toutes les lignes contenant la chaîne de caractères TITRE sont visualisées (comme si elles étaient dans un tampon). Il est dès lors possible de les modifier avant de repasser en mode normal.

```
uc CNTEXT
```

Passage en mode mise à jour de contexte. Toutes les lignes du contexte dont l'identificateur est CNTEXT sont visualisées (comme si elles étaient dans un tampon). Il est dès lors possible de les modifier avant de repasser en mode normal.

4.62 DEMANDE UH

La demande UH (UnHide) permet de revenir au mode de visualisation normal pour l'ensemble de lignes spécifié, ce qui supprime l'effet d'une demande H (Hide) précédente pour ces lignes.

Paramètres de la demande :

paramètre adresse : ensemble de lignes
valeur implicite : la ligne courante
paramètres supplémentaires : aucun

Exemples :

```
uh 40:50
```

Les lignes 40 à 50, précédemment cachées, sont visualisées.

```
uh *
```

Retour au mode de visualisation normal pour toutes les lignes du tampon précédemment cachées.

4.63 DEMANDE UL

La demande UL (Update Line) permet de mettre à jour un ensemble de lignes spécifié. Les lignes à traiter sont visualisées l'une après l'autre avec une zone de mise à jour.

La mise à jour peut s'effectuer en mode local, les lignes étant réintroduites dans le tampon en appuyant sur la touche de transmission (TRANS).

Paramètres de la demande :

```
paramètre adresse           : ensemble de lignes
  valeur implicite         : la ligne courante
paramètres supplémentaires : aucun
```

Cette demande fait passer l'utilisateur en mode mise à jour.

Dans ce mode, FSE affiche une zone de mise à jour constituée de huit champs : 4 pour la mise à jour (dont la longueur est de 70 caractères), et 4 marges (de 7 caractères) protégées.

Le contenu de la ligne à mettre à jour est affiché dans les 4 champs de mise à jour (le premier champ va de la colonne 1 à 70, le deuxième de la colonne 71 à 140, le troisième de la colonne 141 à 210 et le quatrième de la colonne 211 à 255).

Les caractères non graphiques sont affichés sous la forme [Xhh, hh étant le code hexadécimal du caractère.

Il est possible de faire tous les changements souhaités quant au contenu de cette ligne mais les espaces de fin de ligne sont conservés, cette demande ne permettant ni d'en supprimer ni d'en rajouter. Les lignes dont la longueur est supérieure à 255 caractères sont tronquées.

Une fois la ligne mise à jour, il est possible :

- de valider les changements effectués en appuyant sur la touche de transmission (TRANS) et en introduisant un espace dans la zone "ctl", ce qui a également pour effet de visualiser la ligne suivante à mettre à jour. Si la ligne qu'on vient de traiter est la dernière ligne à modifier, la session de mise à jour prend fin.
- d'introduire une barre oblique (/) dans la zone de gestion (ctl), auquel cas le contenu de la zone de mise à jour est ignoré et la session de mise à jour prend fin.

Après exécution de la demande, la ligne courante est la dernière ligne mise à jour. Si aucune ligne n'a été mise à jour, la ligne courante est la ligne courante précédente.

Demandes "Ligne"

Remarque importante :

Il existe deux façons de sortir du mode mise à jour :

- (1) Si l'on est au niveau de la dernière ligne à mettre à jour (qui est la dernière ligne spécifiée par le paramètre adresse), il suffit d'appuyer sur la touche de transmission (TRANS).
- (2) Sinon, placer le curseur dans la zone "ctl" de l'écran (en appuyant sur la touche ORIGINE - HOME) puis introduire une barre oblique (/). Dans ce cas, les modifications apportées à la ligne courante sont ignorées, celles portant sur les lignes précédentes étant, par contre, prises en compte.

Exemple :

ul 200

Passage en mode mise à jour pour la ligne 200.

Avant l'exécution de la demande, la grille suivante s'affiche :

```
b=MAIN m=LIB: DEM DAT > ln=2470
ctf  _____

100 THIS IS RECORD NO 100
110 THIS IS RECORD NO 110
120 THIS IS RECORD NO 120
130 THIS IS RECORD NO 130
140 THIS IS RECORD NO 140
150 this is record no 150
160 this is record no 160
170 this is record no 170
180 this is record no 180
190 this is record no 190
200 this is record no 200
>>>> 210 THIS IS RECORD NO 210
220 THIS IS RECORD NO 220
230 THIS IS RECORD NO 230
240 THIS IS RECORD NO 240
250 THIS IS RECORD NO 250
260 THIS IS RECORD NO 260
270 THIS IS RECORD NO 270
280 THIS IS RECORD NO 280
290 THIS IS RECORD NO 290

requests _____
_____
```

Une fois la demande exécutée, la grille se présente comme suit :

b=MAIN m=LIB: DEM DAT > In=2470	ctl	_____
100 THIS IS RECORD NO	100	
110 THIS IS RECORD NO	110	
120 THIS IS RECORD NO	120	
130 THIS IS RECORD NO	130	
140 THIS IS RECORD NO	140	
150 THIS IS RECORD NO	150	
160 THIS IS RECORD NO	160	
170 THIS IS RECORD NO	170	
>>> 180 THIS IS RECORD NO	180	
190 THIS IS RECORD NO	190	
200 THIS IS RECORD NO	200	
210 THIS IS RECORD NO	210	
220 THIS IS RECORD NO	220	
230 THIS IS RECORD NO	230	
240 THIS IS RECORD NO	240	
250 THIS IS RECORD NO	250	
260 THIS IS RECORD NO	260	
270 THIS IS RECORD NO	270	
280 THIS IS RECORD NO	280	
290 THIS IS RECORD NO	290	
requests UL 200	_____	_____

4.64 DEMANDE ULF

La demande ULF (Update Line Formatted) permet de mettre à jour des lignes en mode formaté, en utilisant une application (fournie par le système) qui correspond au langage utilisé.

Les formats de langages standards sont acceptés. (Voir au chapitre 3 le tableau de ces langages).

Les lignes à mettre à jour sont visualisées l'une après l'autre.

Les lignes peuvent être mises à jour en mode local, et être réintroduites ensuite dans l'unité en appuyant sur la touche de transmission (TRANS).

Paramètres de la demande :

```
paramètre adresse          : ensemble de lignes
    valeur implicite       : la ligne courante
paramètres supplémentaires : aucun
```

Après exécution de cette demande, la ligne courante est la dernière ligne de l'ensemble de lignes mis à jour.

Si l'exécution de la demande est interrompue, la ligne courante est la dernière ligne affichée en bas de l'écran.

Les caractères non graphiques ne peuvent pas être traités par cette demande. Il faut utiliser la demande UL (Update Line).

Remarque importante :

Pour sortir du mode formaté, placer le curseur dans la zone "ctl" de l'écran (en appuyant sur la touche ORIGINE - HOME) et introduire une barre oblique (/).

Exemple :

```
ulf 30:40
```

Passage en mode formaté. Il est maintenant possible de mettre à jour les lignes 30 à 40.

4.65 DEMANDE UP

La demande UP (UPper) permet de mettre en majuscules un ensemble de lignes spécifié.

Paramètres de la demande :

```
paramètre adresse      : ensemble de lignes
valeur implicite      : la ligne courante
paramètres supplémentaires : aucun
```

Après exécution de la demande, la ligne courante est la dernière ligne modifiée.

Exemple :

```
up 150:200
```

Les lignes 150 à 200 doivent être mises en majuscules. Avant l'exécution de la demande, la grille affichée est la suivante :

```
b=MAIN m=LIB: DEM DAT > ln=2470
120  THIS IS RECORD NO 120
130  THIS IS RECORD NO 130
140  THIS IS RECORD NO 140
150  THIS IS RECORD NO 150
160  THIS IS RECORD NO 160
170  THIS IS RECORD NO 170
180  THIS IS RECORD NO 180
190  THIS IS RECORD NO 190
     THIS IS RECORD NO 200 BEING UPDATED
     _____
     _____
210  THIS IS RECORD NO 210
220  THIS IS RECORD NO 220
230  THIS IS RECORD NO 230
240  THIS IS RECORD NO 240
250  THIS IS RECORD NO 250
260  THIS IS RECORD NO 260
270  THIS IS RECORD NO 270
280  THIS IS RECORD NO 280
290  THIS IS RECORD NO 290
requests _____
_____
```

Demandes "Ligne"

Après l'exécution de la demande, la grille affichée se présente comme suit :

b=MAIN m=LIB: DEM DAT > ln=2470	ctl	_____
100	this is record no	100
110	this is record no	110
120	this is record no	120
130	this is record no	130
140	this is record no	140
150	this is record no	150
160	this is record no	160
170	this is record no	170
180	this is record no	180
190	this is record no	190
200	this is record no	200
>>>> 210	this is record no	210
220	this is record no	220
230	this is record no	230
240	this is record no	240
250	this is record no	250
260	this is record no	260
270	this is record no	270
280	this is record no	280
290	this is record no	290
requests UP 150: 200	_____	_____

4.66 DEMANDE USTP

La demande USTP (UnSTeP) met fin au mode pas à pas.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

```
ustp
```

Sortie du mode pas à pas.

4.67 DEMANDE ?

La demande ? (Help), permet de visualiser la liste complète des demandes "ligne".

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

?

La liste des demandes "ligne" est visualisée.

4.68 DEMANDE =

La demande = permet de visualiser une nouvelle fois, dans la zone "request", la dernière demande "ligne" qui a été exécutée.

Noter les points suivants :

- Les demandes 'R' et '=' elles-mêmes ne sont jamais considérées comme dernière demande exécutée.
- Les macro-demandes sont considérées comme des demandes exécutées. Par conséquent, si la dernière demande était le nom d'un tampon d'exécution 'x', la demande 'R' ré-affichera le contenu du tampon 'x'.

Paramètres de la demande :

paramètre adresse : aucun
paramètres supplémentaires : aucun

Après exécution de la demande, la ligne courante est inchangée.

Exemple :

```
dem _1; dem_2; = ;
```

Les demandes 1 et 2 sont exécutées et la demande 2 s'affiche à nouveau pour validation ou modification.

4.69 DEMANDES > ET <

Les demandes > et < permettent le défilement vers le bas et vers le haut respectivement, d'un ou plusieurs écrans.

Paramètres de la demande :

```
paramètre nombre           : entier positif
  valeur implicite         : 1
paramètres supplémentaires : aucun
```

Le paramètre "nombre" indique de combien d'écrans il faut remonter ou descendre.

Les demandes > et < définissent le sens implicite pour les demandes de défilement qui suivent.

Un appui sur la touche de transmission (TRANS) avec une zone "request" vide provoque le passage à l'écran suivant ou précédent, selon le sens implicite. Si le dernier déplacement résultait d'une demande >, la valeur implicite est le passage à l'écran suivant. Si le dernier déplacement résultait d'une demande <, la valeur implicite est le passage à l'écran précédent.

Exemples :

```
> 10
```

Il y a déplacement de dix écrans vers le bas.

```
< 5
```

Il y a déplacement de 5 écrans vers le haut.

4.70 DEMANDE <NOM DE TAMPON>

Cette demande permet l'exécution d'une séquence de demandes enregistrées dans un tampon auxiliaire.

N'importe quel tampon peut être "exécuté" (son contenu est traité comme s'il s'agissait du contenu de la zone "request").

Il est possible d'appeler une macro en introduisant simplement son nom à la place du nom d'une demande, et celui-ci peut être suivi de n'importe quel paramètre souhaité.

Paramètres de la demande :

```
premier paramètre   : param1
    valeur implicite : zone vide
deuxième paramètre  : param2
    valeur implicite : zone vide
troisième paramètre : param3
    valeur implicite : zone vide
```

Le premier, deuxième ou troisième paramètre (même si la zone correspondante est vide) remplace, à l'intérieur du tampon, la séquence d'échappement [1, [2 ou [3, respectivement. Une séquence d'échappement [n ne peut être qu'une valeur complète de paramètre. Par conséquent, la syntaxe du paramètre macro-demande doit respecter rigoureusement la syntaxe du paramètre correspondant dans les demandes paramétrées du tampon.

En cas de passation d'un paramètre pour lequel il n'y a pas de séquence d'échappement correspondante dans le tampon, aucun message d'erreur n'est émis et le paramètre est ignoré.

Exemples :

```
DRAW_TAB
```

Les demandes de modifications FSE contenues dans le tampon DRAW_TAB sont exécutées.

```
ECHANGE quatre cinq 40
```

Dans le tampon ECHANGE, "[1" est remplacé par "quatre", "[2" par "cinq", et "[3" par "40". Ensuite, les demandes de modifications FSE contenues dans le tampon sont exécutées.

L'exemple précédent appelle quelques remarques supplémentaires :

Demandes "Ligne"

Si le tampon ECHANGE contient :

```
S [3 [1 #;  
S [3 [2 [1;  
S [3 # [2;
```

il est possible de l'appeler de différentes façons :

```
ECHANGE Jean John *;
```

pour remplacer Jean par John dans tout le tampon,

ou bien

```
ECHANGE Jean John;
```

pour obtenir ce remplacement dans la ligne courante uniquement, étant donné que le troisième paramètre est laissé en blanc (la valeur implicite du paramètre adresse étant utilisée pour chaque demande S contenue dans le tampon ECHANGE).

- Remarque :**
1. Les premier et deuxième paramètres doivent respecter la syntaxe décrite pour l'ancienne chaîne de caractères et pour la nouvelle dans la demande S. Il faut veiller à ne pas utiliser les séquences spéciales qui n'ont pas la même signification dans l'ancienne et la nouvelle chaîne de caractères, par exemple les séquences /a*/ , /^/ ou /&/ .
 2. Le troisième paramètre doit respecter la syntaxe du paramètre adresse. Ce doit être un numéro de ligne, sinon la deuxième et la troisième substitution ne porteront pas sur les mêmes lignes que la première substitution. On peut aussi utiliser un identificateur de contexte pour définir l'ensemble de lignes touchées. Dans ce cas, le contenu du tampon ECHANGE est le suivant :

```
ASG [3 wctxt;  
S wctxt [1 #;  
S wctxt [2 [1;  
S wctxt # [2;
```


5. Mode Ecran

5.1 PRESENTATION

L'intérêt du mode écran est d'ajouter la fonction mise à jour de FSE à l'ensemble des demandes déjà disponibles en mode ligne dans la zone "request".

Pour passer en mode écran (SE - Screen Edit), la demande SE doit être introduite dans la zone "request".

L'appel de SE est interprété comme une tentative de modification du tampon. Si l'unité de bibliothèque associée au tampon a déjà été réservée par quelqu'un d'autre, un message d'erreur est généré et la demande est rejetée.

Pour sortir du mode écran (SE), il suffit d'introduire LE dans la zone "request".

5.2 DESCRIPTION DES GRILLES

Lors du passage en mode écran, la grille suivante s'affiche :

```

b=MAIN mb=LIB: DEM DAT no context > no bounds 2470 lines      ctl_____
—— 101  this is record nb 101
—— 102  this is record nb 102
—— 103  this is record nb 103
—— 104  this is record nb 104
—— 105  this is record nb 105
—— 106  this is record nb 106
—— 107  this is record nb 107
—— 108  this is record nb 108
—— 109  this is record nb 109
—— 110  this is record nb 110
—— 111  this is record nb 111
—— 112  this is record nb 112
—— 113  this is record nb 113
—— 114  this is record nb 114
—— 115  this is record nb 115
—— 116  this is record nb 116
—— 117  this is record nb 117
—— 118  this is record nb 118
—— 119  this is record nb 119
—— 120                                120

requests_____
_____

```

La présentation de la grille n'a pas changé, mais en mode écran, les zones texte et numéro de ligne ne sont pas protégées et sont donc accessibles.

La zone de gestion (ctl) est elle aussi accessible. L'introduction d'une barre oblique (/) dans cette zone a pour effet de tout annuler dans la zone texte et de visualiser à nouveau la grille non modifiée.

5.3 OPTIONS DU MODE ECRAN

En mode écran, trois types d'options sont disponibles pour le traitement de texte :

- (1) la mise à jour directe du texte visualisé, par des manipulations du curseur.
- (2) l'introduction de demandes dans les zones "numéro de ligne" (ou marges), qui font 8 caractères. Ces demandes permettent :
 - d'insérer ou d'ajouter un ou plusieurs articles à partir de la console ou à partir d'un autre tampon,
 - de copier ou de transférer un ou plusieurs articles dans un autre tampon,
 - de supprimer un ou plusieurs articles,
 - de supprimer toutes les occurrences d'un caractère donné dans un article,
 - de mettre à jour un article avec une zone de mise à jour,
 - de couper ou de concaténer des articles,
 - de convertir des majuscules en minuscules et vice versa,
 - de définir des sauts,
 - de cacher un ou plusieurs articles ou, au contraire, de les rendre visualisables, s'ils ont été cachés,.
- (3) l'utilisation de la totalité des demandes disponibles en mode ligne (LE), particulièrement celles qui ne peuvent pas être introduites dans les zones "numéro de ligne", comme le changement de ligne courante ou le sauvegarde d'unité.

La séquence d'exécution est la suivante :

- la mise à jour du texte visualisé et des demandes "écran" (dans la marge de gauche) est effectuée ligne par ligne, de la première ligne à la dernière de la zone texte affichée,
- les demandes spécifiées dans la zone "requests" sont traitées.

Texte Mise à jour

5.3.1 Mise à jour du texte

Il est possible de modifier le texte affiché soit en introduisant les modifications, qui "écrasent" le texte antérieur, soit en utilisant les touches d'insertion ou suppression (effacement) du terminal. Ne pas oublier de valider les modifications en appuyant sur la touche de transmission (TRANS).

Pour annuler toutes les modifications introduites depuis le dernier appui sur la touche TRANS, si l'on ne veut pas qu'elles soient appliquées à l'unité, il suffit d'introduire une barre oblique (/) dans la zone "ctl". Pour annuler uniquement les modifications relatives à une ligne, la barre oblique doit être introduite dans la zone "numéro de ligne" correspondante.

Comme il n'existe pas de façon simple de visualiser les espaces en fin de ligne, le seul moyen d'en supprimer ou d'en ajouter est d'utiliser la demande S (Substitution).

Les lignes contenant des caractères non graphiques ont fait l'objet d'une attention particulière. Dans ce mode, ces lignes s'affichent en vidéo inverse avec des points qui remplacent en général ces caractères, selon le caractère en question et le type du terminal. Aucune modification de texte n'est effectuée sur ces lignes afin d'éviter le remplacement intempestif de ces caractères par des points. Néanmoins, il est possible d'ajouter ou de supprimer des caractères non graphiques dans une ligne en utilisant la demande UL de mise à jour de ligne.

5.3.2 Demandes "écran"

Les demandes "écran" sont introduites dans la zone "numéro de ligne" des articles auxquels elles s'appliquent.

Un article peut correspondre à plusieurs lignes à l'écran mais une demande "écran" pour cet article ne peut être introduite que dans la première zone "numéro de ligne".

Mode Ecran

Exemple:

```
b=MAIN mb=LIB: DEM DAT no contex t > no bounds 2470 lines      ctl_____
_____ 101  this is record nb      101
_____ 102  this is record nb      102
_____ 103  this is record nb      103
_____ 104  this is record nb      104
D_____ 105  this record whose length is greater than 70 characters
and lower than 210 characters, extends over three
screen lines
_____ 106  this is record nb      106
_____ 107  this is record nb      107
_____ 108  this is record nb      108
_____ 109  this is record nb      109
_____ 110  this is record nb      110
_____ 111  this is record nb      111
_____ 112  this is record nb      112
_____ 113  this is record nb      113
_____ 114  this is record nb      114
_____ 115  this is record nb      115
_____ 116  this is record nb      116
_____ 117  this is record nb      117
_____ 118  this is record nb      118

requests _____
_____
```

La demande "écran" D (suppression) s'applique à l'article portant le numéro 105 qui s'affiche sur trois lignes. Il est interdit d'introduire des demandes dans les zones "numéro de ligne" des deux dernières lignes de cet article.

5.3.2.1 SYNTAXE

La syntaxe de la demande "écran" est la suivante :

```
nom_demande [paramètre]
```

Le nom de la demande et son paramètre (s'il y en a un) doivent être séparés par un espace.

Le paramètre doit appartenir à l'un des types suivants :

- paramètre de type caractère : un seul caractère entre deux apostrophes (') ou guillemets (").
- paramètre de type chaîne : une chaîne de caractères entre deux apostrophes (') ou guillemets (").
- paramètre de type nom : même convention que pour les noms de tampon. Une forme abrégée est acceptée à condition qu'il n'y ait pas d'ambiguïté.

Exemples :

```
ul
dc 't'
sp 'les'
k EXJ
```

Remarque : Veiller à séparer les demandes (avec leur paramètre) du numéro de ligne, notamment si ce dernier comporte plus de deux chiffres. Introduire un espace après le dernier paramètre de la demande.

5.3.2.2 DEMANDES "ECRAN" PORTANT SUR UNE SEULE LIGNE

Ces demandes, qui ne s'appliquent qu'à une seule ligne, sont les suivantes :

I, A	Insertion ou adjonction d'une ou plusieurs lignes. Ces demandes sont semblables à celles du mode ligne. Noter que la zone texte en dehors de la zone de saisie est protégée pendant la session de saisie. Il faut sortir de la session de saisie pour reprendre la mise à jour du texte.
UL	Mise à jour de cette ligne. La différence qui existe entre cette demande et l'option mise à jour de texte du mode écran est que la demande UL permet l'agrandissement de l'article mis à jour et le traitement des caractères non graphiques (éventuels). Si l'article mis à jour contient des caractères non graphiques, ils sont affichés sous la forme [Xhh (hh étant le code hexadécimal du caractère). Il est possible d'introduire jusqu'à 255 caractères non graphiques.
DC car	Suppression de toutes les occurrences du caractère spécifié par le paramètre car. Ce paramètre est obligatoire et doit être mis entre apostrophes (') ou guillemets ("). (Ex: DC 'c', DC "").
D	Suppression de cet article.
SP chaîne	Coupeure de l'article chaque fois que la chaîne spécifiée est détectée. Le paramètre chaîne est facultatif (la valeur implicite est '%') et doit être mis entre apostrophes (') ou guillemets (").
CC	Concaténation de cet article avec le précédent. Noter que les deux articles peuvent être non contigus pendant une session de mise à jour de contexte.
/	Annulation de toutes les modifications effectuées au niveau de cet article.
K nom_tampon	Copie de cet article à la fin du tampon spécifié par nom_tampon. Ce nom peut être abrégé à condition qu'il n'y ait pas d'ambiguïté.
M nom_tampon	Transfert de cet article à la fin du tampon spécifié par nom_tampon. Ce nom peut être abrégé à condition qu'il n'y ait pas d'ambiguïté.
H	Occultation de cet article.
UH	Visualisation des articles cachés.
UP	Conversion de cet article en majuscules.

Mode Ecran

LO	Conversion de cet article en minuscules.
IF,AF	Insertion ou adjonction d'un ou plusieurs articles en mode formaté.
ULF	Mise à jour de la ligne article en mode formaté. Cette demande n'est pas applicable aux articles contenant des caractères non graphiques.
PF	Impression de cet article en mode formaté.
FC caractère_saut	Enregistrement d'un caractère de commande de saut dans l'en-tête SSF de cette ligne. Ce caractère peut être 0, 1, +, - ou un espace. Il doit être mis entre apostrophes (') ou guillemets ("). (Ex : FC '1').
AB nom_tampon	Adjonction, après cet article, des lignes contenues dans le tampon spécifié.
IB nom_tampon	Insertion, avant cet article, des lignes contenues dans le tampon spécifié.

5.3.2.3 DEMANDES PORTANT SUR UN BLOC DE LIGNES

Deux demandes, spécifiées l'une au niveau de la première ligne, l'autre au niveau de la dernière ligne, permettent de traiter un bloc de lignes, et évite ainsi d'introduire une demande pour chaque ligne touchée par l'opération.

Il s'agit, en fait, d'une extension des demandes "écran" qui portent sur une seule ligne.

L'extension d'une demande X est une double demande : BX (Begin_X) et EX (End_X).

A noter que l'article défini par BX doit précéder l'article défini par EX. D'autre part, il peut y avoir plusieurs écrans entre la demande BX et la demande EX et inversement.

Il est possible de faire n'importe quelle opération entre les deux : la demande X n'est exécutée qu'une fois EX détecté. Après quoi, elle porte sur le résultat obtenu entre le moment où X est appelé le moment où X est exécuté. Il est recommandé de clore une demande bloc le plus tôt possible. Toute demande bloc laissée ouverte sera signalée par un message approprié dans la zone en tête.

Pour invalider une demande bloc, il suffit d'introduire une barre oblique (/) devant l'article début.

Les demandes bloc suivantes peuvent être utilisées :

BD, ED	tous les articles de BD (Begin_Delete) à ED (End_Delete) sont supprimés.
BM [nom_tampon], EM [nom_tampon]	tous les articles de BM (Begin_Move) à EM (End_Move) sont transférés dans le tampon spécifié. Celui-ci n'a besoin d'être spécifié qu'une seule fois; une erreur est signalée si deux noms de tampon différents sont fournis pour BM et EM.
BK [nom_tampon], EK [nom_tampon]	tous les articles de BK (Begin_copy) à EK (End_copy) sont copiés dans le tampon spécifié. Celui-ci n'a besoin d'être spécifié qu'une seule fois; une erreur est signalée si deux noms de tampon différents sont fournis pour BK et EK.

BH, EH tous les articles de BH (Begin_Hide) à EH (End_Hide) sont cachés.
 BUP, EUP tous les articles de BUP (Begin_UP) à EUP (End_UP) sont convertis en majuscules.
 BLO, ELO: tous les articles de BLO (Begin_LO) à ELO (End_LO) sont convertis en minuscules.

Exemple :

```

b=MAIN mb=LIB: DEM DAT no context > no bounds 2470 lines   ctl   _____
UF  _____ 101  this is record nb 101
   _____ 102  this is record nb 102
   _____ 103  this is record nb 103
   _____ 104  this is record nb 104
BM BUF          this is record nb 105
   _____ 106  this is record nb 106
EM BUF          this is record nb 107
BD _____ 108  this is record nb 108
   _____ 109  this is record nb 109
ED _____ 110  this is record nb 110
   _____ 111  this is record nb 111
BH _____ 112  this is record nb 112
   _____ 113  this is record nb 113
   _____ 114  this is record nb 114
EH _____ 115  this is record nb 115
   _____ 116  this is record nb 116
BK BUF1        this is record nb 117
   _____ 118  this is record nb 118
   _____ 119  this is record nb 119
   _____ 120  this is record nb 120

requests _____
    
```

Les demandes introduites ont pour effet, après appui sur la touche de transmission (TRANS) :

- de convertir l'article 101 en caractères majuscules,
- de transférer les articles 105, 106, 107 dans le tampon BUF,
- de supprimer les articles 108, 109, 110,
- de cacher les articles 112, 113, 114, 115,
- d'indiquer que le début du bloc d'articles à copier dans BUF1 est l'article 117. La demande de fin de bloc correspondante qui n'est pas spécifiée sur cette grille devra l'être sur une grille suivante.

Mode Ecran

Le résultat de l'exécution de ces demandes sur le texte visualisé est le suivant :

```
PENDING REQUESTS : BK AT LN 117                                ctl _____
____ 96      this is record nb          96
____ 97      this is record nb          97
____ 98      this is record nb          98
____ 99      this is record nb          99
____ 100     this is record nb          100
____ 101     THIS IS RECORD NB          101
____ 102     this is record nb          102
____ 103     this is record nb          103
____ 104     this is record nb          104
____ 111     this is record nb          111
           4 hidden line(s)
____ 116     this is record nb          116
____ 117     this is record nb          117
____ 118     this is record nb          118
____ 119     this is record nb          119
____ 120     this is record nb          120
____ 121     this is record nb          121
____ 122     this is record nb          122
____ 123     this is record nb          123
____ 124     this is record nb          124
requests _____
_____
```

A noter qu'un message est affiché dans la zone en-tête pour indiquer qu'il faudra introduire la demande de fin de bloc correspondante dès que la dernière ligne du bloc à copier apparaîtra à l'écran.

5.3.3 Demandes "ligne"

Il est possible, en mode écran, d'utiliser la totalité des demandes du mode ligne décrites au chapitre 4.

Pour ce faire, il faut d'abord positionner le curseur sur la zone "requests".

Attention, la ligne courante a pu être modifiée par la fonction de mise à jour de texte ou par exécution d'une demande "écran".

Après exécution des demandes "ligne", la ligne courante est la dernière ligne modifiée que ce soit en mise à jour directe à l'écran ou par l'exécution d'une demande "écran".

