

SOLAR

EDIT16

Éditeur de texte conversationnel

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

ADDITIF A LA NOTICE

EDIT16

MANUEL DE REFERENCE

JANVIER 1983

Additif n° 1 164 095 00 036 05 / FR
concernant la notice n° 1 164 095 00 036 04 / FR

EDIT 16

MANUEL DE REFERENCE 1.164.095.00.036.04 / FR
ADDITIF 1.164.095.00.036.05 / FR

Date : JANVIER 1983

Cet additif modifie le Manuel édité en JUILLET 1981 - il correspond à la livraison d'EDIT 16 : indice IE 07 : JANVIER 1983.

Les modifications suivantes doivent être faites.

Page 3.1

Commande EDIT , description de la syntaxe, nom de catalogue
A la suite du paragraphe décrivant PW,

AJOUTER : "le nom de catalogue par défaut, sous système BOS-D ou
en mode Train de Travaux sous MPES, est << ".

Page 5.20

Directive H (hexadécimal) ; description du fonctionnement
A la suite du paragraphe 1,

AJOUTER : Egalement, dans les directives présentant le contenu d'une
ligne, chaque apparition du code hexadécimal défini entraîne
l'impression du caractère imprimable correspondant ; ceci
tant que la correspondance établie par la directive H n'aura
pas été annulée par une autre directive H ou par une fin de
session".

Page 12.1

Mise en oeuvre, configuration nécessaire

REEMPLACER : "15 Kmots" par : "16 Kmots"

Page 13.9

MESSAGES D'ERREURS FATALES : signification du message ERE 1

REEMPLACER le paragraphe par :

"Chaînage des lignes incorrect : espace virtuel désorganisé
à la suite d'une erreur de matériel, non identifiée".

EDIT16

Gamme : SOLAR

Systèmes : BOS-D, TSM, MPES

Objet : Description de l'éditeur de texte interactif
EDIT16, utilisation et mise en oeuvre.

Date d'édition : JUILLET 1981

AVERTISSEMENT :

Ce manuel remplace Le précédent manuel 1 164 095 00 036 03 / FR
(correspondant à EDIT16 IE05) pour la présentation des nouveautés propres à
EDIT16 VERSION 6).

Deux fonctions nouvelles sont disponibles :

1. L'édition d'articles de fichier indexé traitée intégralement dans le
nouveau chapitre : "Accès aux articles de fichier indexé". Celui-ci
présente des compléments aux paragraphes suivants :

- Commande EDIT § 3.2
- Directives G § 7.1
- W § 7.2
- M § 9.3.1
- F § 3.3.1
- S § 3.4.1

- Sommaire des directives et messages en annexe

2. La directive H, qui permet le traitement de caractères non imprimables,
fait l'objet d'un nouveau paragraphe dans le chapitre : "Directives de
traitement de texte".

- Directive H(hexa) § 5.6

1	PRESENTATION GENERALE.	1.1
2	NOTIONS PRINCIPALES	2.1
2.1	SYNOPTIQUE DES ENVIRONNEMENTS EDIT16	2.4
3	SESSION EDIT16	3.1
3.1	ACTIVATION DE L'EDITEUR	3.1
3.2	DEBUT DE SESSION : COMMANDE EDIT.	3.1
3.3	FIN DE SESSION	3.3
3.3.1	DIRECTIVE F (FILE)	3.3
3.3.2	DIRECTIVE Q (QUIT)	3.4
3.4	SAUVEGARDE EN COURS DE SESSION	3.4
3.4.1	DIRECTIVE S (SAVE)	3.4
3.5	EXEMPLE DE SESSION	3.6
3.6	UNITES SYMBOLIQUES D'EDIT16	3.6
3.7	UNITE SUPPORT DU FICHER EDITE	3.7
3.8	UTILISATION EN MODE TRAIN DE TRAVAUX : BATCH.	3.7
4	GENERALITES SUR LES DIRECTIVES.	4.1
4.1	DESCRIPTION GENERALE	4.1
4.2	SYNTAXE	4.1
4.2.1	NOTATION UTILISEE dans ce manuel.	4.2
4.3	REPETITION DE DIRECTIVE : DIRECTIVE A (AGAIN)	4.2
4.4	MACRO-DIRECTIVE	4.3
5	DIRECTIVES DE TRAITEMENT DE TEXTE	5.1
5.1	MODIFICATION DU SEPARATEUR DE CHAINE	5.2
5.1.1	DIRECTIVE "/"	5.2
5.2	MOUVEMENT ET IMPRESSION DU POINTEUR DE LIGNE	5.2
5.2.1	DIRECTIVE N (NEXT)	5.2
5.2.2	DIRECTIVE U (UP)	5.4
5.2.3	DIRECTIVE T (TOP)	5.5
5.2.4	DIRECTIVE B (BOTTOM)	5.6
5.2.5	DIRECTIVE L (LINE)	5.6
5.2.6	DIRECTIVE O (ORDER)	5.6
5.2.7	DIRECTIVE J (JUMP)	5.7
5.2.8	SYNOPTIQUE DES MOUVEMENTS DU POINTEUR DE LIGNE	5.9

5.3	SUPPRESSION - INSERTION - REMPLACEMENT DE LIGNES	5.10
5.3.1	DIRECTIVE D (DELETE)	5.10
5.3.2	DIRECTIVE I (INSERT)	5.10
5.3.2.1	INSERTION IMMEDIATE	5.10
5.3.2.2	PASSAGE A L'ENVIRONNEMENT INPUT :	5.11
5.3.3	DIRECTIVE R (REPLACE)	5.12
5.4	INFORMATIONS SUR LE CONTENU DE LA LIGNE	5.13
5.4.1	DIRECTIVE P (PRINT)	5.13
5.4.2	DIRECTIVE K (RECHERCHE COLONNES DE CHAINE)	5.14
5.4.2.1	K AVEC CHAINE	5.14
5.4.2.2	K AVEC CHAINE VIDE	5.15
5.4.2.3	K SANS PARAMETRE	5.15
5.4.3	DIRECTIVE X (EXTRACTION CHAINE)	5.16
5.5	MODIFICATION DU CONTENU DE LA LIGNE	5.17
5.5.1	DIRECTIVE C (CHANGE)	5.17
5.5.2	DIRECTIVE Y (INSERTION COLONNE)	5.19
5.6	TRAITEMENT DE CARACTERES NON IMPRIMABLES	5.20
5.6.1	DIRECTIVE H (HEXADECIMAL)	5.20
6	CREATION DE TEXTE	6.1
6.1	ENVIRONNEMENT INPUT	6.1
6.2	CARACTERES ADMISSIBLES	6.2
6.3	MODE TEXTE	6.2
6.4	CREATION EN MODE LECTEUR OU DEPUIS UN FICHIER	6.2
6.5	CREATION DE TEXTE PAR TABULATION ET PAR CHAMPS	6.3
6.5.1	DIRECTIVE E (ESPACEMENT)	6.3
6.5.2	DIRECTIVE > (TABULATION)	6.4
6.5.3	TABULATION	6.4
6.5.4	CHAMPS	6.4
6.5.5	DIRECTIVE E7 (VISUALISATION DES CHAMPS)	6.5
6.5.6	CREATION DE LIGNE PAR CHAMPS	6.5
6.6	CREATION DE TEXTE PAR ZONE	6.6
7	ACCES AUX FICHIERS ET UNITES EXTERNES	7.1
7.1	DIRECTIVE G (GET)	7.1
7.2	DIRECTIVE W (WRITE)	7.2
7.3	IMPRESSION DE LIGNES SUR L'UNITE D'IMPRESSION	7.3
7.3.1	DIRECTIVE P (PRINT) : MODE IMPRIMANTE	7.3

7.4	CREATION DE LIGNES DEPUIS UNE UNITE DE LECTURE CARTES	7.4
7.4.1	DIRECTIVES I (INPUT) ET R (REPLACE) EN MODE LECTEUR	7.4
7.5	SCHEMA UNITES SYMBOLIQUES ET FICHIERS	7.6
8	DEFINITION ET UTILISATION DE ZONE	8.1
8.1	DEFINITION DE ZONE : DIRECTIVE Z	8.1
8.1.1	DEFINITION PROPREMENT DITE	8.1
8.1.2	ANNULATION DE ZONE	8.2
8.1.3	VISUALISATION DE ZONE	8.2
8.1.4	EQUIVALENCE ZONE CHAMP	8.2
8.2	UTILISATION DE ZONE PAR DIRECTIVES	8.3
8.2.1	Directive C	8.3
8.2.2	Directives N et U	8.3
8.2.3	Directive I	8.4
8.2.4	Directive R	8.4
8.2.5	Directive K	8.4
8.2.6	Directive X	8.4
8.2.7	Directive Y	8.5
8.2.8	Directive P	8.5
8.2.9	Directive W	8.5
8.2.10	Directive G	8.5
9	MACRO-DIRECTIVES	9.1
9.1	GENERALITES :	9.1
9.1.1	Programmation de MACRO-DIRECTIVES	9.2
9.1.2	Utilisation de MACRO-DIRECTIVES	9.2
9.2	PROGRAMMATION DE MACRO-DIRECTIVES	9.3
9.2.1	INSTRUCTION DE DEFINITION	9.3
9.2.2	INSTRUCTIONS DE FIN DE DEFINITION	9.3
9.2.3	CORPS DE MACRO-DIRECTIVE	9.4
9.2.4	INSTRUCTIONS DE MACRO-DIRECTIVES	9.4
9.2.5	UTILISATION DES PARAMETRES	9.4
9.2.6	DEFINITION DE VARIABLES : INSTRUCTION * V	9.5
9.2.7	UTILISATION DES VARIABLES	9.6
9.2.8	EXPRESSIONS ARITHMETIQUES	9.6
9.2.9	EXPRESSIONS CHAINE	9.6
9.2.10	RUPTURE DE SEQUENCE	9.7
9.2.10.1	INSTRUCTION *SKIP	9.7

9.2.10.2	INSTRUCTION *IF SKIP	9.7
9.2.10.5	INSTRUCTION *KILL	9.8
9.2.11	INSTRUCTIONS DE DIALOGUE	9.8
9.2.11.1	INSTRUCTION *PAINT	9.8
9.2.11.2	INSTRUCTION *LPRINT	9.9
9.2.11.3	INSTRUCTION *REPLY	9.9
9.2.11.4	INSTRUCTION *T	9.9
9.2.12	EXECUTION DE DIRECTIVES - COMPTE RENDU	9.10
9.2.12.1	VARIABLE VOO	9.10
9.2.12.2	PARAMETRE PO	9.10
9.2.13	EXEMPLE DE MACRO-DIRECTIVE	9.10
9.2.14	CAS DE LA DIRECTIVE A (AGAIN)	9.11
9.2.15	UTILISATION DE CARACTERES NON-IMPRIMABLES	9.11
9.3	CREATION ET APPEL DE MACRO-DIRECTIVES	9.12
9.3.1	DIRECTIVE M (MACRO)	9.12
9.3.2	DIRECTIVE %	9.14
9.4	COMPLEMENT : CLASSEMENT DES MACRO-DEFINITIONS	9.14
9.5	QUELQUES EXEMPLES DE MACRO-DIRECTIVES	9.15
10	PARTICULARITES DE FONCTIONNEMENT	10.1
10.1	AUTRES COMMANDES DE L'EDITEUR	10.1
10.1.1	COMMANDES GO ET STOP	10.1
10.1.1.1	Utilisation de l'appel opérateur (BREAK)	10.1
10.1.2	COMMANDE SPACE	10.2
10.2	FICHIERS DE TRAVAIL DE L'EDITEUR	10.3
10.3	FICHIERS A ENREGISTREMENTS DE LONGUEUR FIXE.	10.4
10.3.1	LONGUEUR 80 CARACTERES.	10.4
10.3.2	LONGUEUR INFERIEURE A 80 CARACTERES	10.5
11	ACCES AUX ARTICLES DE FICHIER INDEXE	11.1
11.1	COMMANDES ET DIRECTIVES	11.1
11.1.1	COMMANDE EDIT (activation d'EDIT16)	11.1
11.1.2	DIRECTIVES S (SAVE) et F (FILE)	11.2
11.1.3	DIRECTIVE W (WRITE)	11.3
11.1.4	DIRECTIVE G (GET)	11.3
11.1.5	DIRECTIVE M (MACRO)	11.3
11.2	PARTICULARITES D'UTILISATION DES FICHIERS INDEXES	11.4
11.2.1	CREATION DU FICHIER INDEXE	11.4

11.2.2	GESTION DU FICHIER INDEXE11.5
11.2.3	COMPLEMENTS ET EXEMPLES11.5
12	MISE EN OEUVRE DE L'EDITEUR12.1
12.1	CONFIGURATION NECESSAIRE12.1
12.2	INTEGRATION DE L'EDITEUR12.1
12.2.1	MODE MAITRE12.1
12.2.2	ADAPTATION DES PARAMETRES.12.1
12.2.3	INTEGRATION SOUS LES SYSTEMES TSM et TSF/MPES12.3
12.2.4	CONTRAINTES FMS12.3
13	ANNEXE13.1
13.1	SOMMAIRE COMMANDES ET DIRECTIVES13.1
13.1.1	COMMANDES D'ACTIVATION13.1
13.1.2	DIRECTIVES DE SAUVEGARDE ET FIN DE SESSION13.1
13.1.3	DIRECTIVES SUR POINTEUR DE LIGNE13.2
13.1.4	DIRECTIVES SUR CONTENU DE LIGNE13.3
13.1.5	DIRECTIVES SUR LIGNES13.4
13.1.6	DIRECTIVES D'ACCES FICHIERS et MACRO DIRECTIVES13.5
13.2	SOMMAIRE ECRITURE DE MACRO-DIRECTIVES13.6
13.3	MESSAGES13.7
13.3.1	MESSAGES D'INFORMATION13.7
13.3.2	MESSAGES D'ANOMALIES13.8
13.3.3	MESSAGES D'ERREURS FATALES : ERE n13.9
13.3.4	MESSAGES D'ERREURS MACRO-DIRECTIVES : ERMnn13.10

1 PRESENTATION GENERALE.

EDIT16 est un éditeur de texte conversationnel. Il permet le traitement de textes, enregistrés dans des fichiers de type séquentiel ou dans des articles de fichier de type indexé. Ces textes sont composés de lignes considérées comme des images de carte de 80 caractères. Il est possible de traiter des textes comprenant jusqu'à environ 10.000 lignes.

EDIT16 apporte des possibilités variées en création et édition de texte.

- Recherches de lignes de chaîne, globales ou limitées.
Déplacements relatifs et absolus
- Chargement de contenu de lignes ou groupes de lignes :
localisation, échange, insertion
- Mise en forme du texte, insertion, suppression, remplacement
- Transfert de lignes avec autres fichiers
- etc.

L'édition est commandée par un dialogue composé de directives et de réponses simples. Le jeu de directives peut être enrichi par l'utilisateur par définition de macro-directives, permettant de réaliser de façon simple des opérations répétitives ou complexes ou de prévoir des utilisations personnalisées.

EDIT16 fonctionne en mode conversationnel avec les systèmes d'exploitation BOS-D, TSM, MPES et en mode train de travaux avec tous les systèmes d'exploitation à disque du SOLAR16.

2 NOTIONS PRINCIPALES

SESSION EDIT16

Une session EDIT16 est l'ensemble des opérations réalisées entre :

- . La demande de création d'un nouveau texte ou d'édition d'un texte existant déjà sur un fichier.
- . La demande finale se traduisant ou non par l'enregistrement effectif des modifications apportées.

Durant une session, les opérations de traitement, sont traitées sur un fichier temporaire (appelé parfois fichier virtuel), image du fichier ou de l'article contenant le texte origine. La technique utilisée permet une réduction du temps de traitement et la conservation du fichier origine jusqu'au moment choisi par l'utilisateur.

DIRECTIVE

Une directive est l'action demandée par l'utilisateur à l'éditeur après ouverture d'une session. Elle est généralement obtenue par frappe au clavier d'un caractère unique ou d'une ligne commençant par un caractère qui est suivi de paramètres : une syntaxe simple doit être respectée.

La réponse de l'éditeur sera l'action demandée sanctionnée parfois par la présentation d'une réponse. En cas d'erreur de syntaxe, la réponse sera l'émission d'un caractère "?" invitant l'utilisateur à émettre une directive correcte. En cas de problème en cours d'exécution l'émission d'un message d'information ou d'erreur peut survenir.

ENVIRONNEMENTS

On appelle environnements les états différents dans lesquels peut se trouver l'éditeur à divers moments d'une session.

- EDIT :

C'est le mode normal de fonctionnement au cours duquel l'éditeur est en attente de directives. Les caractères présentés sur l'organe d'entrée, en général le clavier, sont considérés comme des directives à exécuter.

- INPUT :

L'entrée de lignes est réalisable dans cet environnement correspondant à la création de lignes nouvelles. Les caractères fournis sont considérés comme du texte.

Il est obtenu par la demande d'édition d'un texte vide ou par une directive I(Input) ou R(Replace). Il se quitte par émission d'une ligne vide.

Si la notion de macro-directives est utilisée deux environnements supplémentaires sont définis. (Voir chapitre spécial Macro-Directives).

- MACRO DEFINITION :

Les caractères fournis sont considérés comme des définitions de

macro-directives.

Obtenu par une directive M(Macro) cet environnement se termine par ligne vide ou fin de fichier.

- MACRO GENERATION :

La génération, résultant de la demande d'exécution d'une macro-directive, fournit des lignes qui sont immédiatement interprétées comme des directives ou des lignes de texte. Obtenu par la directive " % " d'appel de macro-directive cet environnement se termine sur une instruction contenue dans la macro-définition.

TEXTE

Le texte est contenu dans un fichier séquentiel ou un article de fichier indexé situé sur un disque géré par FMS.

Il est constitué de lignes de longueur variable (jusqu'à 80 caractères).

Il est créé soit au cours d'une session EDIT16 en environnement INPUT soit par un autre moyen.

Sa taille se situe entre un minimum d'une ligne et un maximum défini, par option, à environ 2500, 5000 ou 10000 lignes. Ses bornes sont la ligne 0, appelée TOP (sommet), ligne vide située avant la première ligne du texte, et la dernière ligne de texte qui est nommée BOTTOM (fond).

LIGNE

La ligne est un ensemble de caractères normalement imprimables, en nombre variable : de 1 à 80.

Sa longueur peut être considérée par l'utilisateur comme étant de 80 caractères : image des 80 colonnes d'une carte perforée. Pour l'éditeur cette longueur est variable car elle est déterminée par le dernier caractère défini pour la ligne. Celui-ci est suivi du caractère de fin de ligne : "RC".

A chaque ligne, sont associés deux numéros :

Le numéro logique

Il correspond à la succession logique des lignes dans le texte, tel que l'a constitué l'utilisateur. Il peut varier pour une ligne donnée en fonction des insertions ou suppressions de lignes dans le texte qui la précède.

Le numéro absolu

IL correspond à la position physique de la ligne dans le fichier de travail de l'éditeur. Pour une ligne donnée il ne varie pas pendant toute la durée d'une session.

Les numéros absolus, identiques en début de session aux numéros logiques sont le reflet de l'ordre logique de création des lignes : une ligne nouvelle reçoit un numéro immédiatement supérieur au dernier numéro attribué. La suppression d'une ligne provoque un vide dans la séquence des numéros absolus. Ce numéro permet le déplacement immédiat vers une ligne (directive J=JUMP).

LIGNE COURANTE

L'éditeur entretient tout au long d'une session un pointeur sur la ligne en cours de traitement : c'est la ligne courante.

Ce pointeur évolue (et en conséquence la ligne courante désigne une ligne différente) quand sont exécutées certaines directives.

La plupart des directives concernent implicitement la ligne courante ou une portion de texte commençant à la ligne courante.

A tout moment le contenu de la ligne courante peut être obtenu par la

directive P(Print), ses limites par la directive K(Key), son numéro logique par L(Line), son numéro absolu par O(Order).

GROUPE DE LIGNES

Certaines directives permettent le traitement d'un sous ensemble du texte. Elles définissent alors un nombre de lignes, en général relativement à la ligne courante, sur lesquels sont réalisés les opérations demandées.

CONTEXTE

Concept important de l'éditeur, c'est la localisation rapide d'une chaîne de caractères identique à une chaîne précisée dans une directive. Elle permet la recherche d'une ligne par contexte, c'est à dire, de celle qui contient cette chaîne et des modifications rapides de l'ensemble ou de parties de texte par substitution de chaînes.

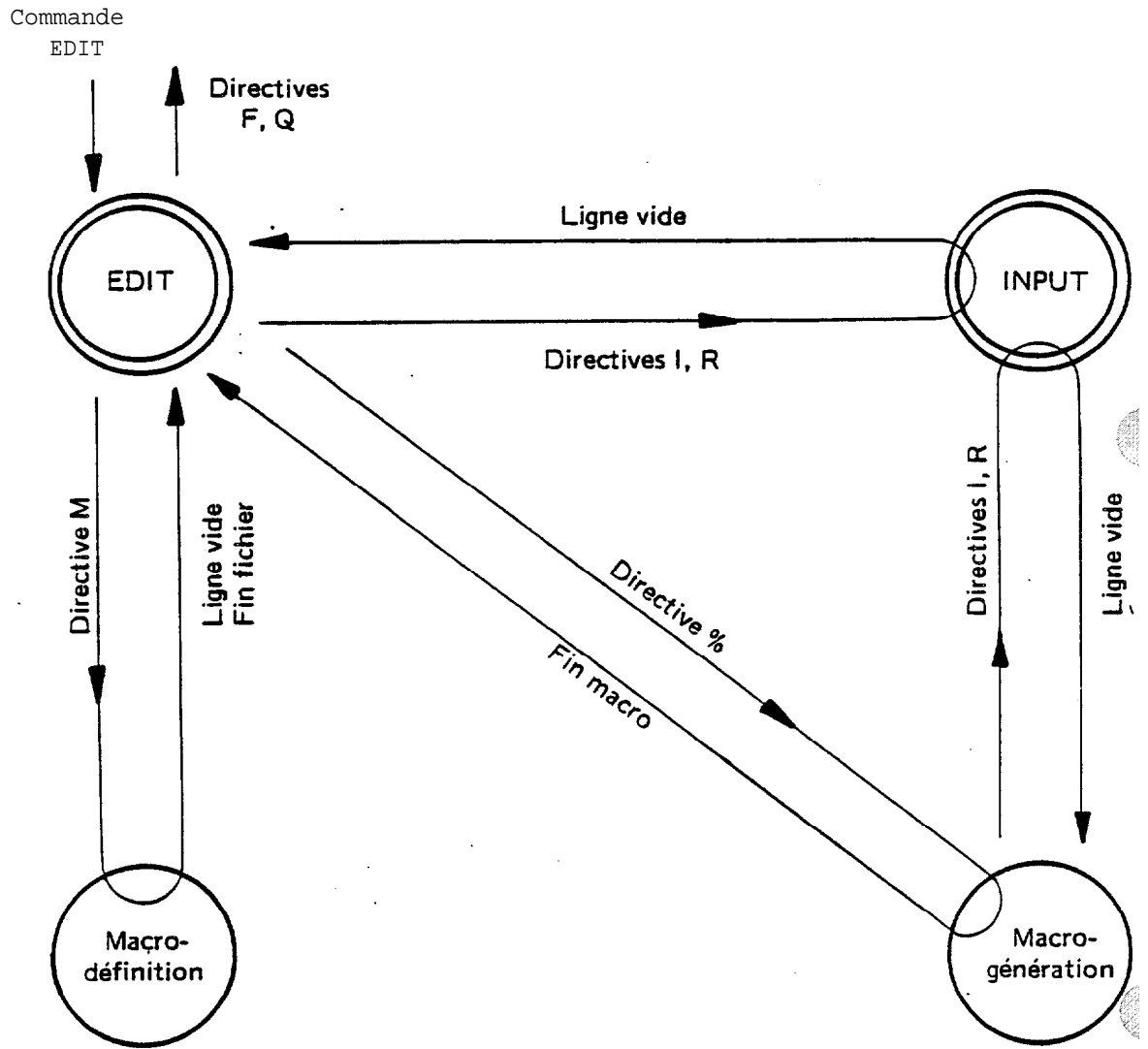
COLONNES

Les caractères ayant dans les diverses lignes du texte la même position relative par rapport au début de ligne constituent une colonne. L'ensemble des Nièmes positions d'un texte constitue la colonne N.

ZONE

Alors que l'édition concerne normalement le contenu d'une ligne dans sa totalité, il est possible de limiter temporairement et dynamiquement la portée de certaines directives à une partie de la ligne appelée ZONE afin de préserver le reste de la ligne ou simplement de ne pas la considérer. La zone, définie par l'utilisateur, modifiable et annulable à tout moment détermine un sous-ensemble vertical du texte exprimé en colonnes de début et de fin, par opposition au groupe de lignes qui définit un sous-ensemble horizontal.

2.1 SYNOPTIQUE DES ENVIRONNEMENTS EDIT16



Changements d'environnements - EDIT16

3 SESSION EDIT16

3.1 ACTIVATION DE L'EDITEUR

L'activation de l'éditeur est réalisée au moyen de la commande :

```
CALL EDIT16
```

Le lancement permettant d'initialiser une session EDIT16 sera ensuite possible par l'utilisation de la commande EDIT.

Dans certains cas, il sera parfois nécessaire de réaliser auparavant le lien entre les unités symboliques définies par l'éditeur, et les unités fonctionnelles de système à l'aide de commandes d'assignation SU-FU.

Lors de son activation, EDIT16 transmet au système

- la commande EDIT pour le lancement d'une édition
- trois autres commandes (SPACE, GO, STOP). Elles correspondent à des utilisations particulières et sont décrites au chapitre PARTICULARITES DE FONCTIONNEMENT.

3.2 DEBUT DE SESSION : COMMANDE EDIT.

BUT :

Début d'une session d'édition

SYNTAXE :

```
EDIT nomfic[-pw] [,Ui|,Di|,Ei][,NS] [T]
```

Nomfic : nom du fichier à éditer (6 caractères maximum)

pw : nom de catalogue du fichier (2 caractères maximum)
(Sous système multiusager, si ce paramètre est absent, le nom par défaut est le n° de compte : 2 derniers caractères de LOGIN).

Ui, Di, Ei : Unité symbolique à laquelle est affectée une unité fonctionnelle disque, ou unité fonctionnelle elle-même, qui est support du fichier (i = 1 à F, sauf D1). Par défaut, l'unité fonctionnelle standard de l'utilisateur (ou celle définie dans la commande JOB de BOS-D).

NS : Non Simultané - l'édition de fichiers de ce type (voir FMS) est possible sur système multiconsole en utilisant ce paramètre. Elle permet que soit

rejetée une demande d'édition simultanée par un autre utilisateur. L'utilisation ultérieure de ce fichier nécessitera la présence du paramètre NS, associé au nom de fichier.

T : Mode TEXTE. La session sera réalisée dans ce mode, les caractères alphabétiques minuscules seront distingués des majuscules. Voir chapitre CREATION DE TEXTE.

Remarque : EDITION D'ARTICLES DE FICHER INDEXE

La syntaxe de la commande EDIT permet la forme "nomartic.nomfic" en lieu et place de la forme "nomfic". Voir chapitre : ACCES AUX ARTICLES DE FICHER INDEXE.

FONCTIONNEMENT :

Le fichier nommé est recherché. 2 cas peuvent se présenter :

- . Le fichier est trouvé : il est alors recopié sous une forme spéciale dans un fichier de travail géré par l'éditeur. (Cette opération provoque, sur un système mono-console un temps d'attente de l'ordre de 15 secondes, pour un texte d'un millier de lignes). L'éditeur se place alors en mode d'attente de directives, et le signale par le message d'information : EDIT :
- . Aucun fichier de ce nom n'est trouvé. L'éditeur en crée un nouveau, se met en attente de lignes ; celles-ci viendront constituer le fichier de travail de l'éditeur. L'éditeur le signale en émettant les messages d'information :

NEW FILE
INPUT :

Si le fichier de travail n'existe pas ou n'a pas été fermé correctement par l'utilisation précédente (absence de directives de fin d'édition), il est automatiquement recréé. Cette opération nécessite une dizaine de secondes.

REMARQUE :

Sous système multi-console les messages EDIT et INPUT sont précédés du message PUBW = i, donnant le nom de catalogue choisi par le système pour le fichier de travail et éventuellement le fichier intermédiaire de l'éditeur.

MESSAGES D'ERREUR

ERE n : Une erreur a été détectée. Voir signification en Annexe

NO SPACE : Il n'est pas possible de définir le fichier de travail : voir commande SPACE au chapitre : PARTICULARITES DE FONCTIONNEMENT

WRONG TYPE Le fichier n'est pas de type séquentiel.

La fin d'une session se réalise par l'émission d'une directive F ou d'une directive Q qui permettent de quitter l'environnement d'édition.

3.3.1 DIRECTIVE F (FILE)

BUT :

Terminer la session en sauvegardant les modifications apportées.

SYNTAXE :

F

Remarque : Pour une utilisation particulière, avec création d'enregistrements de longueur fixe, une autre forme existe : voir chapitre PARTICULARITES.

FONCTIONNEMENT

La directive F (FILE) fait quitter l'environnement EDIT. Elle réalise auparavant l'opération de recopie dans le fichier, nommé au moment de la commande EDIT, des modifications qui viennent d'être apportées. Après impression, pour mémoire, du nom de fichier concerné (et, si on est sous système multi-console, du numéro de PUBW des fichiers de travail) l'éditeur rend le contrôle au superviseur

Remarque : EDITION D'ARTICLE DE FICHER INDEXE

La syntaxe est identique - le fonctionnement est adapté au type du fichier - voir le chapitre : ACCES AUX ARTICLES DE FICHER INDEXE.

MESSAGES

- 1. Nomfic-pw

END

Message d'information signalant la fin de session, après recopie correcte du fichier modifié. Cette opération peut nécessiter une dizaine de secondes pour un fichier de 1000 lignes.

- 2. Nomfic-pw

ERE n

Une erreur a été détectée pendant la recopie du fichier (voir signification du message d'erreur en annexe). Le fichier origine tel qu'il a été la dernière fois sauvegardé, mais sans les modifications de l'actuelle session peut être restitué. Voir chapitre PARTICULARITES. On peut ensuite recommencer l'édition.

Remarque :

Pour la sauvegarde du fichier sous forme d'enregistrement 80 caractères, voir le chapitre PARTICULARITES.

3.3.2 DIRECTIVE Q (QUIT)

BUT :

Terminer la session en ignorant les modifications éventuellement demandées.

SYNTAXE :

Q

FONCTIONNEMENT :

La directive Q (QUIT) fait quitter l'environnement EDIT. Les modifications apportées au cours de cette session ne sont pas recopiées dans le fichier origine : celui-ci reste tel qu'il a été la dernière fois sauvegardé par la directive F (File) au cours d'une précédente session ou une directive S (Save) dans la présente session. Le contrôle est rendu au superviseur. Une nouvelle commande EDIT peut être émise.

MESSAGE :

END fin de la session

3.4 SAUVEGARDE EN COURS DE SESSION

Il est possible de demander avant la fin de session une sauvegarde des modifications déjà réalisées dans le fichier origine : directive S, ou dans un nouveau fichier (voir chapitre 7 : directive W)

3.4.1 DIRECTIVE S (SAVE)

BUT :

Sauvegarde des modifications apportées et reprise de l'édition.

SYNTAXE :

S

FONCTIONNEMENT :

Les modifications qui ont été demandées jusqu'alors sont intégrées dans le fichier source nommé lors de la commande EDIT. Après impression pour mémoire du nom du fichier édité, l'éditeur se place en mode EDIT, en attente de nouvelles directives. La ligne courante reste la même qu'avant la directive.

Remarque : EDITION D'ARTICLES DE FICHIER INDEXE

La syntaxe est identique - le fonctionnement est adapté au type du fichier - voir le chapitre : ACCES AUX ARTICLES DE FICHIER INDEXE.

MESSAGE :

Nomfic-pw

EDIT : Recopie correcte des modifications dans le fichier.

ERE n

Erreur dans la recopie du fichier. (voir directive F et signification de l'erreur en annexe).

Remarque :

- Cette directive ne devrait être utilisée qu'en cas d'édition de longue durée si l'on craint d'avoir à recommencer toute l'édition en cas d'incident (défaut matériel, coupure secteur...).
- La directive F réalise les mêmes opérations qu'une directive S suivie d'une directive Q.

3.5 EXEMPLE DE SESSION

Une session EDIT16 se présente généralement sous la forme suivante :

```
CALL EDIT16
EDIT NOUVEL-MM
  NEW FILE          *
  INPUT :          *
  ....
  .... (Lignes de texte)
  ....
  EDIT :           *
  ..              *
  .. (directives utilisateur)
  ..
  F (directive file)

  NOUVEL-MM       *
  END              *
EDIT ANCIEN-A6
  EDIT :          *
  ..
  ..
  ..
  F

  ANCIEN-AB       *
  END              *
. . .
```

Les lignes marquées d'une étoile sont des messages émis par EDIT16.

3.6 UNITES SYMBOLIQUES D'EDIT16.

Les opérations d'échange d'entrée-sortie de l'éditeur sont effectuées sur des unités symboliques. Après activation de l'éditeur il est possible de modifier l'affectation standard prévue par le système en utilisant les commandes d'assignation d'unités symboliques à unités fonctionnelles. Les unités symboliques sont au nombre de 5. Seules les 2 premières sont nécessaires dans tous les cas.

- PC : pour l'entrée des directives et des lignes en mode "input". (normalement le clavier du terminal de dialogue)
- EL : pour l'impression normale et la sortie de messages (normalement l'imprimante ou l'écran du terminal de dialogue).
- L0 : pour l'impression en mode "imprimante" : directive P : (normalement l'imprimante rapide LP)
- SI : pour l'entrée de lignes en mode "input" sur lecteur : directives 1 : et & : (normalement l'unité de lecture de carte CR)

EC : pour l'entrée de réponse à l'instruction de dialogue REPLY à l'intérieur d'une macro-directive.

Un SYNOPTIQUE de l'ensemble des entrées sorties et de l'environnement fichier est présenté au chapitre : ACCES AUX FICHIERS et UNITES EXTERNES.

3.7 UNITE SUPPORT DU FICHIER EDITE

Au moment de la commande d'activation EDIT, le fichier nommé est recherché sur l'unité standard du système, ou celle redéfinie par la commande JOB de BOS-D. Cependant la syntaxe de la commande EDIT permet de préciser que la recherche devra être réalisée sur :

- une unité symbolique de la forme Ui qui aura été préalablement affectée à une unité fonctionnelle disque gérée par FMS
- une unité fonctionnelle disque de la forme Di ou Ei. L'unité fonctionnelle D1 non gérée par FMS ne peut être acceptée.

EXEMPLE : édition du texte contenu dans le fichier TEXTE-SY de la FU D3 au moyen de la console de visualisation F1.

- CALL EDIT16
- PC F1
- EL F1
- U1 D3
- EDIT TEXTE-SY,U1

3.8 UTILISATION EN MODE TRAIN DE TRAVAUX : BATCH.

EDIT16 peut être utilisé en mode BATCH. Dans ce cas, les unités PC, EL et SI sont affectées au lecteur de cartes, les unités EL et LO à l'imprimante rapide. Pour conserver la trace de toutes les opérations réalisées sur le texte, l'éditeur recopie alors sur l'imprimante les directives émises et les lignes entrées.

Afin de rendre la liste plus lisible, les directives recopiées sont précédées des caractères "-" et ">".

EXEMPLE

Lecture en mode train de travaux de la directive suivante sur carte

N /TOTO

Sur la liste, apparaissent les lignes

-> N/TOTO/	- Directive recopiée
TOTO - TUTU	- Ligne imprimée

4 GENERALITES SUR LES DIRECTIVES.

4.1 DESCRIPTION GENERALE

Une directive est constituée d'une ligne, parfois limitée à un seul caractère. Cette ligne est terminée par le caractère "RC" obtenu par la frappe de la touche Retour ou Return si entrée au clavier. Pour les claviers qui le permettent, les caractères seront fournis indifféremment en majuscules ou en minuscules.

4.2 SYNTAXE

<directive> [<zone>] [<mode>] [paramètre] ["RC"]

<directive>	nom d'une directive limitée à un seul caractère
<zone>	un caractère qui, s'il est utilisé, sera placé immédiatement derrière la lettre de directive (sans espace) il peut être : - non utilisé - Le caractère Z : pour limiter l'effet de la directive à une ZONE prédéfinie - il peut être interdit pour certaines directives : voir chapitre ZONES - un caractère de A à Y pour limiter l'entrée de caractères à un CHAMP d'une ligne en cours de création : il est réservé aux directives I et R : voir CHAMPS au chapitre CREATION DE TEXTE.
<mode>	spécifie le mode de fonctionnement. IL peut être omis : c'est le mode normal.
"."	mode muet : la réponse normalement fournie par la directive (ligne courante, numéro de colonne etc...) n'est pas imprimée
""	mode sans message : le message fourni par l'éditeur quand l'exécution de la directive entraîne la rencontre de certaines conditions n'est pas imprimée. Utilisation conseillée uniquement dans le corps de macro-directives.
""	mode muet et sans message : c'est l'addition des 2 modes précédents. Conseillé uniquement dans le corps de macro-directives.
""	mode imprimante : la réponse normalement fournie par la directive est dirigée sur l'unité symbolique LO normalement assignée à une imprimante rapide. Dans le cas des directives I (insert) et R (replace) le caractère "." désigne le mode lecteur dans lequel les lignes à insérer sont lues sur

l'unité symbolique SI normalement affectée à un lecteur de carte.

<paramètre> certaines directives nécessitent des paramètres obligatoires ou acceptent des paramètres facultatifs. Ils sont séparés par le caractère espace " ".

"RC" le caractère Retour Chariot correspondant à la frappe de la touche RETURN viendra terminer toute directive entrée au clavier. On ne le présentera pas dans la syntaxe des directives. Il n'est pas nécessaire en cas d'entrée par cartes.

Règles d'écriture :

Il n'est pas nécessaire de mettre un espace entre la directive et ses paramètres.

Le caractère "flèche arrière" "<-" provoque l'annulation de tous les caractères qui précèdent.

Le caractère "flèche en l'air "↑" provoque l'annulation du caractère valide qui précède.

4.2.1 NOTATION UTILISEE dans ce manuel.

La syntaxe des directives est décrite en utilisant les signes suivants :

[...] : toute entité entre crochets est facultative

(...|...) les entités entre parenthèses sont exclusives

/ : représente le caractère délimiteur de chaîne de caractères éventuellement modifié par l'utilisateur ; le dernier / n'est pas obligatoire si la directive est frappée au clavier

> : représente le caractère de tabulation éventuellement modifié par l'utilisateur

Les caractères ".", ",", ";" et ":" spécifiant le mode de fonctionnement peuvent suivre la plupart des directives. Ils n'ont pas en général, été représentés dans la syntaxe, sauf si leur présence nécessitait des commentaires complémentaires.

Le caractère de zone (ou de champ) ne sera pas en général représenté dans la syntaxe des directives pour lesquelles il est autorisé.

4.3 REPETITION DE DIRECTIVE : DIRECTIVE A (AGAIN)

But :

Répéter la dernière directive émise

Syntaxe :

A[n]

n est un entier indiquant le nombre de répétitions, par défaut 1.

Fonction :

Répéter une ou plusieurs fois l'exécution de la directive précédente. Exécution non récursive, c'est-à-dire qu'une succession de AGAIN s'adresse à la dernière directive exécutée autre que A (Again).

Exemples :

1. N
 A49

Ceci est équivalent à N50. Directive N réalisée 1 fois puis 49 autres fois

2. N
 A29

A20 Ceci est équivalent à N50. Directive N réalisée 1 fois puis 29 fois puis 20 fois

4.4 MACRO-DIRECTIVE

On appelle macro-directive un ensemble d'opérations que l'utilisateur peut définir à l'aide de macro définitions et utiliser ultérieurement. Il peut ainsi réaliser des successions de directives, utilisables dans les mêmes conditions qu'une directive et adapté à son besoin. Voir le chapitre MACRO-DIRECTIVES.

5 DIRECTIVES DE TRAITEMENT DE TEXTE

IL existe plusieurs catégories de directives permettant de réaliser un traitement de texte en cours d'édition.

- Modification du séparateur de chaîne

/ SEPARATOR

- Mouvement et Impression du Pointeur de ligne

U UP

N NEXT

T TOP

B BOTTOM

L LINE

O ORDER

J JUMP

- Suppression Insertion Remplacement de ligne

D DELETE

I INSERT

R REPLACE

- Information sur le contenu de la ligne

P PRINT

K KEY

X EXTRACT

- Modification du contenu de la ligne

C CHANGE

Y INSERT COLONNE

- Traitement de caractères non imprimables

H HEXADECIMAL

- Les directives portant sur le contenu de la ligne, ou de plusieurs lignes peuvent être utilisées avec les limitations dues à une définition de zone. On se reportera pour cette utilisation particulière au chapitre ZONE.



5.1.1 DIRECTIVE "/"

Syntaxe :

/ car

Car : caractère quelconque non numérique

Fonctionnement :

La directive / permet de remplacer le caractère standard de séparation de chaîne pour les directives qui seront par la suite émises. Ainsi le caractère "/" pourra être utilisé dans les chaînes considérées.

On évitera d'utiliser les caractères de mode, soit . , : et :

Le retour à la situation d'origine se fera par utilisation de la directive sous la forme "//"

5.2 MOUVEMENT ET IMPRESSION DU POINTEUR DE LIGNE

5.2.1 DIRECTIVE N (NEXT)

But :

Déplacer en avant le pointeur de ligne courante

Syntaxe : (forme normale)

N [n|/chaîne/]

n	est un entier qui indique le nombre de lignes dont il faut faire avancer le pointeur. Par défaut, le nombre est pris égal à 1.
/	est le caractère qui délimite la chaîne de caractères. Il n'est pas nécessaire en fin de chaîne si la directive est entrée au clavier
chaîne	est une chaîne de caractères à chercher dans le texte

Fonctionnement :

1. La directive fait progresser le pointeur de n positions. Si n vaut 0 ou n'est pas spécifié, le pointeur est positionné sur la ligne suivante. Si la fin de texte est trouvée avant que le pointeur ait pu avancer de n

positions, il est positionné sur la dernière ligne du texte.

- n suivi d'une chaîne de caractères entre séparateurs provoque l'analyse des caractères de chaque ligne pour trouver la première occurrence de la chaîne. La recherche commence sur la première ligne suivant la ligne courante et continue jusqu'à ce que la chaîne soit trouvée ou sinon jusqu'à la fin du texte. Si la chaîne est trouvée, le pointeur est positionné sur la ligne qui la contient. Si la chaîne n'est pas trouvée, le pointeur est positionné sur la dernière ligne de texte.

Réponse :

La ligne trouvée est imprimée.

Message :

EOF La fin du texte a été trouvée par la directive. Le pointeur est alors positionné sur la dernière ligne.

Exemples :

- | | |
|-----------|---|
| N5 | Le pointeur avance de 5 lignes, et la nouvelle ligne courante est imprimée. |
| N | Le pointeur avance de 1 ligne et la ligne est imprimée. |
| N/FORMAT/ | La première ligne contenant le mot FORMAT à partir de la ligne courante est imprimée, et le pointeur est positionné sur elle. |
| N.3 | Le pointeur avance de 3 lignes et la nouvelle ligne n'est pas imprimée. |

Remarque : ZONE

Si la directive mentionne une zone définie, la recherche de la chaîne sera limitée à la zone : voir chapitre spécial

Syntaxe : (forme étendue) :

N (*|+) /chaîne 1/chaîne 2/---/chaîne n/

chaîne 1,, chaîne n sont des chaînes de caractères à chercher dans le texte.

l représente un "et" sur les chaînes de caractères.

+

représente un "ou" sur les chaînes de caractères.

Fonctionnement :

La directive positionne le pointeur sur la première ligne du fichier vérifiant la condition donnée.

1. Condition * :

Une ligne est solution si L'ENSEMBLE des chaînes de caractères citées appartiennent à la ligne.

2. Condition + :

Une ligne est solution si AU MOINS UNE chaîne de caractères parmi celles citées appartient à la ligne.
Le nombre de chaînes de caractères n'est pas limité.

Réponse :

La ligne trouvée.

Message :

EOF

5.2.2 DIRECTIVE U (UP)

But :

Déplacer en arrière le pointeur de ligne courante

Syntaxe : (forme normale)

U [n|/chaîne/]

n	est un entier qui indique le nombre de lignes dont il faut faire reculer le pointeur. Par défaut, le nombre est pris égal à 1.
/	est le séparateur qui délimite la chaîne de caractères. Il n'est pas nécessaire en fin de chaîne, si la directive est entrée au clavier.
chaîne	est un groupe quelconque de caractères à chercher dans le texte

Fonctionnement :

1. La directive recule le pointeur de n positions.

Si n = 0 ou non spécifié, le pointeur est placé sur la ligne précédente.

Si le début du texte est trouvé avant que le pointeur ait pu reculer de n positions, le pointeur est positionné sur la ligne 0 (équivalent à la requête TOP).

2. U suivi d'une chaîne de caractères entre séparateurs provoque l'analyse des caractères de chaque ligne pour trouver la première occurrence de la chaîne.

La recherche commence sur la première ligne précédant la ligne courante et continue jusqu'à ce que la chaîne soit trouvée ou sinon jusqu'au début du texte.

Si la chaîne n'est pas trouvée, le pointeur est positionné sur la ligne 0.

Réponse :

La ligne trouvée est imprimée.

Message :

TOP

Le début du texte a été trouvé par la requête. Le pointeur est positionné sur la ligne 0.

Exemples :

- | | |
|------------|---|
| U 5 | Le pointeur recule de 5 lignes et la nouvelle ligne est imprimée. |
| U | Le pointeur recule de 1 ligne et la ligne est imprimée. |
| U/ FORMAT/ | La première ligne contenant le mot FORMAT à partir de la ligne courante est imprimée, et le pointeur est positionné sur elle. |
| U.3 | Le pointeur recule de 3 lignes et la ligne n'est pas imprimée. |

Syntaxe : (forme étendue)

La directive U possède une forme étendue, semblable à celle de la directive N (Next), permettant de rechercher une ligne à laquelle appartient un ensemble de chaînes de caractères ou l'une parmi plusieurs chaînes de caractères. Voir la directive N.

Remarque : ZONE

Comme la directive N, la directive U est utilisable avec la notion de zone.

5.2.3 DIRECTIVE T (TOP)

Syntaxe :

T

But :

TOP positionne le pointeur au début du texte, c'est à dire sur la ligne 0. La ligne 0 est une ligne vide qui est insérée par l'éditeur devant la première ligne du texte, afin de permettre l'insertion avant le début du texte.

5.2.4 DIRECTIVE B (BOTTOM)

Syntaxe :

B

But :

BOTTOM positionne le pointeur sur la dernière ligne du texte, et imprime cette ligne.

5.2.5 DIRECTIVE L (LINE)

Syntaxe :

L

But :

Cette directive imprime le NUMERO LOGIQUE de la ligne courante. Après TOP, on imprime 0.

Réponse :

Le numéro logique de la ligne.

5.2.6 DIRECTIVE O (ORDER)

Syntaxe :

O

But :

Cette directive imprime le NUMERO ABSOLU de la ligne courante. Pour les lignes initiales du texte, cette directive est équivalente à la directive LINE.

Réponse :

Le numéro absolu de la ligne.



But :

Positionner le pointeur sur une ligne désignée par son numéro absolu.

Syntaxe :

J [n]

n indique le NUMERO ABSOLU de la ligne sur laquelle on veut positionner le pointeur. Par défaut le numéro est pris égal à 0.

But :

1. Si n vaut 0 ou n'est pas spécifié, cette directive est équivalente à la directive TOP.
2. Si le fichier n'a pas été modifié (pas de directive INSERT, REPLACE, DELETE ou GET), la directive :

J n

est équivalente aux directives

T

N n

Rappels :

1. Le numéro absolu d'une ligne reste invariable quelles que soient les insertions et suppressions effectuées sur le fichier.
2. La suppression d'une ligne a pour effet la disparition du numéro absolu correspondant.

Réponse :

La ligne trouvée est imprimée.

Message :

NO LINE

La ligne demandée n'existe pas (numéro non encore affecté ou disparu).

Exemples :

J 1500	Impression de la ligne
D	Effacement de la ligne
J 1500	

NO LINE Ligne disparue

Sur un fichier de 1000 Lignes.

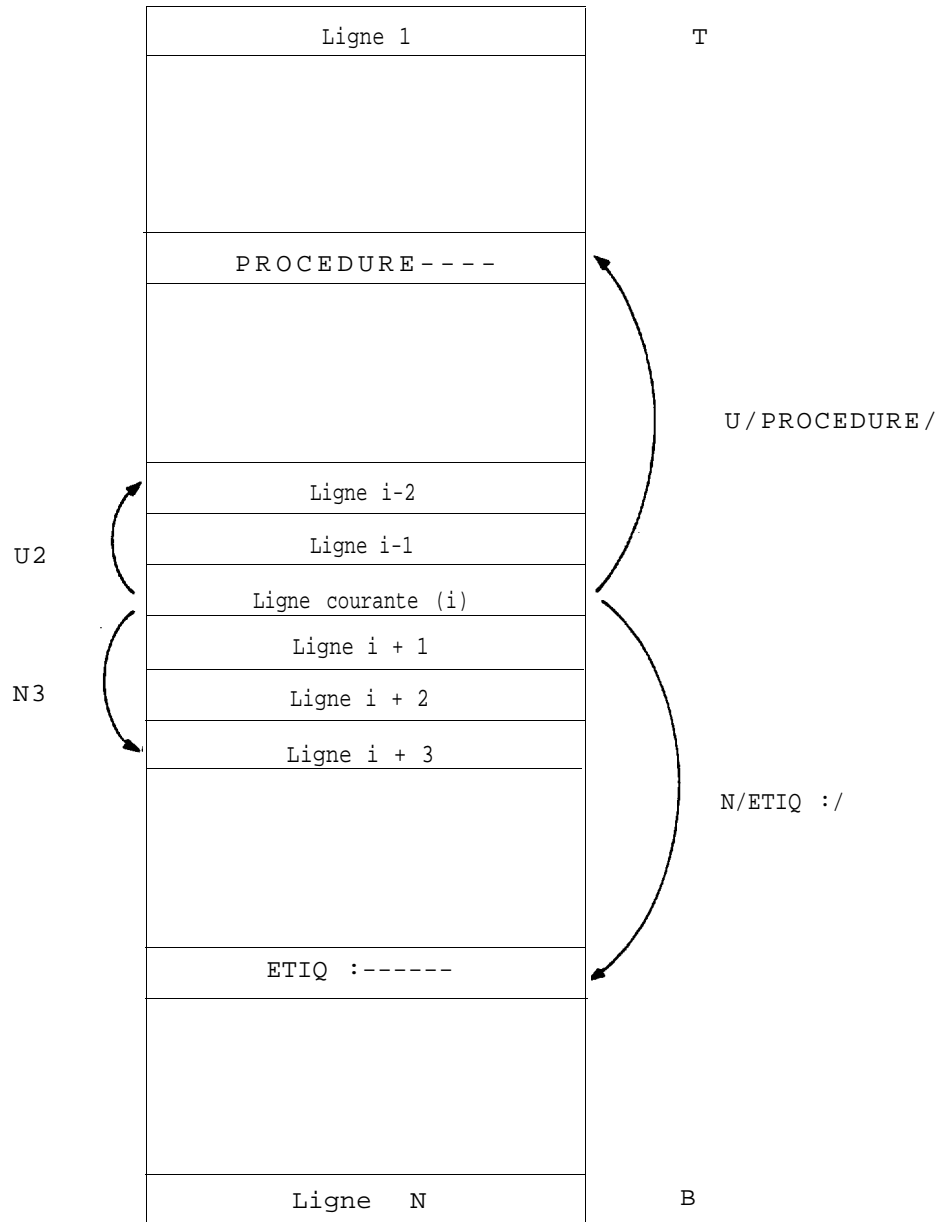
J 1001

NO LINE Ligne pas encore définie

Remarques :

1. La directive J est beaucoup plus performante que la directive N (Next).
2. La directive J fait perdre le numéro logique de la ligne courante : la valeur rendue par L est alors 0 (zéro). Le numéro logique ne sera récupéré qu'à la suite d'une directive T (Top).
3. Si la ligne demandée par la directive J est inexistante, car non créée ou supprimée, le message NO LINE apparaît et le numéro absolu, valeur rendue par la directive O devient 0 (zéro). On se trouve ainsi positionné en début de texte comme après une directive T (Top).

5.2.8 SYNOPTIQUE DES MOUVEMENTS DU POINTEUR DE LIGNE



5.3 SUPPRESSION - INSERTION - REMPLACEMENT DE LIGNES

5.3.1 DIRECTIVE D (DELETE)

But : Supprimer une ou plusieurs lignes

Syntaxe :

D [n]

n : nombre de lignes à supprimer. Par défaut, le nombre est pris égal à 1.

Fonctionnement :

La directive détruit n lignes à partir de la ligne courante incluse. Après exécution, le pointeur est positionné après la dernière ligne effacée.

Si n n'est pas spécifié ou 1, seule la ligne courante est effacée.

Message :

EOF

La fin de fichier a été trouvée par la directive, le pointeur est positionné sur la dernière ligne donc sur la ligne précédant la ligne courante avant exécution de la directive.

Exemples :

D Suppression de la ligne courante.

D10 Suppression de la ligne courante et des 9 lignes suivantes.

5.3.2 DIRECTIVE I (INSERT)

But :

La directive comporte deux formes. La première permet l'insertion immédiate d'une ligne sans quitter l'environnement EDIT. La deuxième forme permet le passage à l'environnement INPUT pour insérer un nombre quelconque de lignes.

5.3.2.1 INSERTION IMMEDIATE

Syntaxe :

I ligne

Ligne est la ligne exacte qui doit être insérée dans le texte. Cette ligne peut contenir des blancs. Elle est séparée du nom de directive par un espace ou un /.

Fonctionnement :

Cette directive insère la ligne dans le texte sans utiliser le mode INPUT.

La ligne est insérée derrière la ligne courante et le pointeur est avancé de 1. La tabulation, l'utilisation de zone et de champs sont possibles. (voir chapitre CREATION DE TEXTE)

La ligne est séparée du nom de la directive seulement par un blanc. Tous les autres blancs font partie de La ligne. Elle peut aussi être séparée du nom de directive par le caractère séparateur de chaîne, en généra "/". (voir le paragraphe CREATION EN MODE TEXTE). Si la ligne est entièrement blanche, il y a passage au mode INPUT (deuxième forme de la directive 1).

On ne peut donc pas insérer directement une ligne blanche.

Après la directive T, ou le message TOP la ligne est insérée en tête du texte, devant la lère ligne.

5.3.2.2 PASSAGE A L'ENVIRONNEMENT INPUT :

Syntaxe :

I

Fonctionnement :

Cette directive permet de passer de l'environnement EDIT à l'environnement INPUT.

Toutes les lignes entrées dans cet environnement sont placées après la ligne courante.

Si une directive INPUT est exécutée après une directive T ou après l'apparition du message TOP, les lignes sont insérées devant la première ligne du fichier.

On quitte l'environnement INPUT (retour à l'environnement EDIT) en entrant une ligne vide ou une carte blanche (mode lecteur).

Le pointeur est positionné sur la dernière ligne entrée.

Remarque

La tabulation, l'utilisation de zone et de champs sont possibles. L'entrée de lignes mode lecteur avec la forme I: possible. Ces types d'utilisation sont détaillés au chapitre CREATION DE TEXTE.

Réponses :

INPUT :

Passage à l'environnement INPUT.

TRUNCATED

La ligne est trop longue supérieure à 80 caractères en tenant compte de la tabulation. Il y a troncation si une ZONE ou un CHAMP a été précisé, la ligne est plus grande que la longueur admissible.

5.3.3 DIRECTIVE R (REPLACE)

But :

Après destruction de la ligne courante, réaliser l'insertion immédiate à la place d'une ligne, sans quitter l'environnement EDIT (1ère forme) ou en passant à l'environnement INPUT (2e forme).

Syntaxes :

R ligne

R

Fonctionnement :

La ligne courante est détruite. Ensuite le fonctionnement est identique à celui de la directive I. (voir cette directive)

Remarque :

La directive R ne peut être assimilée à la directive D suivie d'une directive I. Dans ce dernier cas, la ligne courante est détruite, la ligne suivante devient ligne courante et c'est après celle-ci qu'est réalisée l'insertion. R est équivalent à D, U puis I.

5.4 INFORMATIONS SUR LE CONTENU DE LA LIGNE

5.4.1 DIRECTIVE P (PRINT)

But :

Imprimer la ligne courante ou plusieurs lignes à partir de la ligne courante.

Syntaxe :

P [n]

n : nombre de lignes à imprimer. Par défaut n = 1

Fonctionnement :

La directive réalise l'impression de n lignes à partir de la ligne courante
- la dernière ligne imprimée devient ligne courante
- en l'absence de paramètre ou si n = 0 ou n = 1, seule la ligne courante est imprimée

Réponse :

Impression de la ou des lignes

Message :

EOF
La fin du texte a été trouvée

Exemple :

P 5
Impression de la ligne courante et des 4 lignes suivantes.

MODE IMPRIMANTE

L'impression sur l'imprimante rapide de l'installation est possible en définissant la directive en mode imprimante avec le caractère ":". Il faudra qu'au préalable l'unité symbolique LO ait été assignée à l'unité fonctionnelle d'impression.

Une forme particulière de la syntaxe est disponible dans ce mode imprimante. Voir chapitre ACCES AUX UNITES EXTERNES.

Remarque : ZONE : voir chapitre ZONES

La directive P peut utiliser une définition de zone : seuls les caractères de la zone sont fournis. Ils apparaissent à leur place réelle.

5.4.2 DIRECTIVE K (RECHERCHE COLONNES DE CHAINE)

But :

Localiser une chaîne de caractères et connaître les limites du texte dans la ligne courante. Cette directive est utilisable sous 3 formes différentes.

5.4.2.1 K AVEC CHAINE

Syntaxe :

K / chaîne [/]

/ : caractère délimiteur de chaîne

Chaîne : 1 caractère ou suite de caractères à rechercher

Fonctionnement :

Visualise le numéro de colonne du 1er caractère de chaque occurrence d'une chaîne (si elle existe) dans la ligne courante. Si plusieurs occurrences dans la ligne, plusieurs numéros sont fournis jusqu'à concurrence de 25. Au delà de 25, ils sont remplacés par "..."

Réponse :

Un ou plusieurs numéros de colonne

Message : END LINE : chaîne non trouvée

Exemple :

K /ABC/

12 65

La chaîne ABC apparaît 2 fois dans la ligne courante ; les caractères de début de chaîne : "A" sont situés en colonne 12 et 65.

Remarque : ZONE :

La chaîne est recherchée à l'intérieur de la zone



Syntaxe :

K /[]

Fonction :

Visualise la fin de la ligne

Réponse :

Le numéro de la première colonne disponible dans la ligne, c'est à dire celle qui suit le dernier caractère utilisé.

5.4.2.3 K SANS PARAMETRE

Syntaxe :

K

Fonction :

Visualise les limites du texte

Réponse :

Les numéros de colonne des premier et dernier caractères différents du blanc dans la ligne courante.

Exemple :

K

3 42

Dans la ligne courante, le texte est compris entre les colonnes 3 et 42, celles-ci comprises.

Remarque :

Dans ces deux dernières formes, la directive ne peut être exécutée avec mention de zone.

But :

Connaître la chaîne située à une position donnée, dans une ou plusieurs lignes.

Syntaxe :

X nc [l] [nl|*]

nc : numéro de colonne avec $1 \leq nc \leq 80$

l : longueur, par défaut : 1 avec $nc + l \leq 81$

nl : nombre de lignes (y compris la ligne courante) sur lesquelles appliquer la directive.

. : toutes les lignes depuis la ligne courante

Fonctionnement :

Les caractères de la chaîne débutant à la colonne nc et de longueur l, sont imprimés. Si plusieurs lignes sont examinées, la dernière ligne devient ligne courante.

Réponse :

- Le ou les caractères situés à la position demandée.
- chaîne vide si les colonnes spécifiées ne contiennent aucun caractère :

Exemple 1 :

P

ALPHA = ALPHA - BETA

X 21 4

LPHA

Les colonnes 21, 22, 23 et 24 contiennent LPHA

Exemple 2 :

K /A

14 18 20 24 29

numéro des colonnes contenant la lettre A

X 18

A

5.5 MODIFICATION DU CONTENU DE LA LIGNE

5.5.1 DIRECTIVE C (CHANGE)

But :

Changer le contenu de la ligne en remplaçant une chaîne par une autre

Syntaxe :

C /chaîne 1/chaîne 2/ [n1|*] [n2|*]

/ est le caractère qui délimite les chaînes de caractères. Il peut être remplacé par "cr" après chaîne 2.

Chaîne 1 est le groupe de caractères à remplacer.

Chaîne 2 est le groupe de caractères qui doit remplacer chaîne 1.

n1 nombre de lignes où l'on doit chercher chaîne 1, y compris la ligne courante.

n2 nombre d'occurrences par ligne de chaîne 1 qu'il faut changer.

• La première * signifie que la recherche doit s'effectuer sur l'ensemble des lignes (jusqu'à EOF). La deuxième * signifie qu'il faut changer toutes les occurrences de chaîne 1 dans la ou les lignes

Fonctionnement :

- La directive remplace l'occurrence de la chaîne 1 par la chaîne 2. Chaîne 1 et chaîne 2 peuvent être de longueurs différentes. La ligne modifiée est alors comprimée ou agrandie.

- Si le paramètre n1 est présent, le remplacement pourra s'effectuer pour l'occurrence de la chaîne 1 recherchée dans n1 lignes depuis la ligne courante

- Si le paramètre n2 est présent, les n2 occurrences de chaîne 1 sont remplacées. S'il n'est pas spécifié seule la première occurrence de la chaîne est remplacée.

- En l'absence de ces 2 paramètres, seule la ligne courante est concernée.

- Si chaîne 1 n'est jamais trouvée, les lignes ne sont pas modifiées. Le pointeur est positionné sur la dernière ligne dans laquelle on a cherché chaîne 1.

- Si chaîne 1 est vide (//), chaîne 2 est insérée devant le début de la ligne.

- Si chaîne 2 est vide, chaîne 1 est effacée.

- Si chaîne 1 et chaîne 2 sont vides, la directive est ineffective.

Réponse :

La ou les lignes changées sont imprimées. Si dans une même ligne il y a plusieurs changements, la ligne est imprimée après chaque changement.

Messages :

EOF

La fin du fichier a été trouvée par la directive. Le pointeur est positionné sur la dernière ligne.

NO CHANGE

Chaîne 1 n'existe pas dans la ligne courante et la directive ne fait intervenir que cette ligne.

TRUNCATED

Chaîne 2 est plus long que chaîne 1. La nouvelle ligne a plus de 80 caractères. Les derniers caractères de la ligne sont perdus.

Exemple :

1. C /ALPHA/DELTA/

Ligne avant :

ALPHA = ALPHA - BETA

Ligne après :

DELTA = ALPHA - BETA

2. C /ALPHA/DELTA/ 1 2

Ligne avant :

ALPHA = ALPHA - BETA

Ligne après :

DELTA = DELTA - BETA

3. C /ALPHA/ALPHA/ *

Il n'y a pas de modifications sur l'ensemble du fichier, mais toutes les lignes contenant ALPHA sont imprimées. Le même résultat est obtenu avec les deux directives.

N /ALPHA/

A 10000

Remarque : ZONE

Si la directive est exécutée avec mention de zone, la recherche de chaîne 1 est réalisée dans la zone exclusivement, la substitution par la chaîne 2 se fera en respectant les limites de la zone. Voir CHAPITRE ZONE.



But :

Insérer une chaîne de caractères dans une ou plusieurs lignes

Syntaxe :

Y [nc] /chaîne[/nl|*]

nc : n° de colonne suivi d'un espace. Par défaut : 1
1<=nc<=80

chaîne : suite de caractères à insérer

nl : nombre de lignes sur lesquelles porte l'insertion ;
par défaut : la ligne courante.

. : toutes les lignes, de la ligne courante jusqu'à EOF

Fonction :

Insère la chaîne à partir de la colonne nc ; les caractères situés en colonne nc et suivantes sont décalés d'autant de positions. La ligne est allongée en conséquence.

Réponse :

Le nouveau contenu de la ligne courante ou des lignes concernées.

Message :

"TRUNCATED" si l'insertion a provoqué une perte de caractères en fin de ligne.

Exemple :

```
P
    ALPHA = ALPHA - BETA
K /-
25
Y 25 /* GAMMA
    ALPHA = ALPHA * GAMMA - BETA
```

Remarque : ZONE

Si la directive est exécutée avec mention de zone, l'insertion sera réalisée en considérant le numéro de colonne comme relatif au début de la zone. L'insertion se fera en respectant les limites de la zone. Voir chapitre ZONE.

5.6 TRAITEMENT DE CARACTERES NON IMPRIMABLES

5.6.1 DIRECTIVE H (HEXADECIMAL)

But :

Affecter le code hexadécimal d'un caractère non imprimable à un caractère valide.

Syntaxe :

H car ['XX]

Car : caractère quelconque imprimable
X : caractère hexadécimal, de 0 à F

Fonctionnement :

1. Un code hexadécimal est donné. Ce code sera désormais utilisé à la place du code ASCII normal à chaque apparition du caractère en entrée de donnée ou dans les directives traitant de chaînes de caractères.
2. Sans code hexadécimal. Cette forme de la directive restitue au caractère nommé son code ASCII normal.

Le nombre de caractères redéfinis à un moment donné est limité à 10.

Message :

MAX 10 DEF

Remarque :

Il est recommandé de ne pas utiliser les codes correspondants à "retour chariot" et "nouvelle ligne", ni les caractères utilisés pour la tabulation et la séparation de chaîne.

6 CREATION DE TEXTE

6.1 ENVIRONNEMENT INPUT

Dès le début d'une session, si le fichier ou l'article nommé dans la commande EDIT n'est pas trouvé, l'éditeur se situe en environnement "création de lignes" dit : "INPUT". Il le signale par les messages

NEW FILE ou NEW ITEM suivi de

INPUT :

Les caractères entrés sont alors considérés comme des données devant constituer les lignes du texte.

Pendant une session, l'éditeur étant en environnement EDIT, pour l'entrée de directives, le passage à l'environnement INPUT est réalisable à n'importe quel moment, par l'utilisateur de la directive I (sans paramètre) qui a pour effet de faire apparaître le message :

INPUT :

Les caractères entrés sont considérés comme éléments de lignes. Ces lignes sont alors insérées derrière la ligne courante quel que soit leur nombre.

Une ligne de donnée est considérée comme terminée quand :

- soit la touche "Return" ou "Retour chariot" est frappée, provoquant le positionnement du curseur en début de ligne suivante.
- soit 80 caractères sont entrés

Les caractères suivants sont considérés comme faisant partie d'une nouvelle ligne.

La fin de la création de lignes, (c'est à dire le passage de l'environnement INPUT à l'environnement EDIT) est réalisé par l'émission d'une ligne vide, c'est à dire par la frappe, en début de ligne, de la touche "Retour chariot". Il n'y a pas, pour cette action, création d'une ligne.

La création du fichier ou de l'article contenant le texte préparé ne sera effective qu'à l'émission d'une directive F(File) ou d'une directive S(Save). Voir le chapitre SESSION EDIT16.



Les caractères entrés au clavier pour constituer des lignes de données peuvent être

- des caractères numériques
- des caractères alphabétiques majuscules ou minuscules
- des caractères spéciaux
- des caractères redéfinis par la directive H

pour autant que le clavier utilisé en apporte la possibilité.

Les caractères alphabétiques minuscules sont acceptés et sont équivalents aux caractères majuscules correspondants. Avant traitement, ils sont convertis en majuscules, sauf dans le cas suivant.

6.3 MODE TEXTE

Dans ce mode, demandé lors de commande EDIT la distinction est faite entre caractères minuscules et majuscules.

- En environnement INPUT, il n'y a pas de conversion avant traitement comme en mode normal.
- En environnement EDIT, les caractères minuscules sont considérés comme différents des caractères majuscules correspondants, si ce sont des caractères de données.

Les directives, utilisant des chaînes de caractères acceptent cette distinction entre caractères majuscules et minuscules. Il n'y a pas en effet, de conversion de minuscules en majuscules pour les caractères de la directive, qui sont émis après le caractère de séparation de chaîne ("/" dans le cas standard). Cette distinction est assurée pour les directives N (next), U (UP), C (change), Y (insertion colonne), K (key) et I (insertion ligne) dès lors qu'elles contiennent des caractères précédés du séparateur de chaîne.

6.4 CREATION EN MODE LECTEUR OU DEPUIS UN FICHIER

La création de lignes à partir d'un lecteur de cartes ou d'un fichier assimilé à cette fonction est réalisée à l'aide des directives I et suivies du caractère ":". La création de lignes depuis un fichier contenant un texte est réalisée à l'aide de la directive G (GET).

Pour ces deux utilisations voir le chapitre : ACCES AUX FICHIERS ET UNITES EXTERNES.

Afin de simplifier l'entrée de caractères devant constituer des lignes, dans le cas où des alignements doivent être respectés, il est possible d'utiliser les moyens décrits ici. Ils permettent d'éviter l'entrée de caractères "espace" dont la frappe peut être fastidieuse et être la source d'erreurs.

La création de lignes par tabulation et par champs est rendue possible par la directive suivante.

6.5.1 DIRECTIVE E (ESPACEMENT)

But :

Définition d'un format pour l'entrée de données. Ce format sera utilisable, à la demande, en environnement INPUT avec les directives I et R pour :

- la création de lignes avec tabulation
- la création de lignes par champs.

Syntaxe :

E [n1 n2 n3 ..nj..] [NT|car]

nj : définissent des numéros de colonne, présentés en ordre ascendant, en nombre inférieur ou égal à 24 avec $1 \leq nj \leq 80$.

NT : non tabulation : ces 2 caractères définissent que le caractère de tabulation sera sans effet en cas d'utilisation par champs.

car : nouveau caractère de tabulation pour remplacer le caractère précédemment défini : équivalent à la directive ">".

(Voir au paragraphe CHAMPS, la forme E7)

Fonctionnement :

- La directive E réalise un découpage de la ligne de 80 caractères en sous-ensembles disjoints qui permettent de réaliser l'entrée de données alignées sur les colonnes définies. Selon 2 méthodes : tabulation et champs.
- Le caractère de tabulation est mis en oeuvre, sauf si explicitement refusé par paramètre NT.
- Des champs sont également automatiquement définis.
- En l'absence de numéro de colonne, la direction E annule l'effet de la précédente directive E et rétablit les conditions normales.

Message :

Erreur de syntaxe

Numéros de colonne en nombre supérieur à 24 ou en ordre non croissant ou de valeur supérieure à 80.

But :

Changer le caractère de tabulation

Syntaxe :

>car

Le caractère spécifié remplace le caractère de tabulation. Ceci permet d'utiliser le caractère ">" comme un caractère normal de donnée.

Le retour à la définition standard du caractère se fera en émettant ">>".

On évitera d'utiliser les caractères de mode, soit . , ; et :

Remarque :

Cette redéfinition du caractère de tabulation peut être également faite par un paramètre de la directive E (voir cette directive).

6.5.3 TABULATION

Cette fonction met en oeuvre un mécanisme comparable à la tabulation d'une machine à écrire.

L'entrée de données est réalisée en insérant chaque fois que le caractère de tabulation est rencontré, le nombre d'espaces nécessaire pour atteindre la prochaine colonne définie par la directive E.

En l'absence de directive E préalable le caractère de tabulation est considéré comme un caractère normal.

Le caractère de tabulation standard est le caractère ">".

Exemple :

```
E 12 20 30
I
INPUT :
>>GOTO>DEBUT ;
>END ;
```

GOTO est en colonne 20
DEBUT en colonne 30
END en colonne 12

6.5.4 CHAMPS

Les numéros de colonne établis par la directive E définissent implicitement des champs.

La ligne est découpée entièrement en champs disjoints qui sont implicitement nommés avec les lettres de l'alphabet de A jusqu'à Y. Ainsi la directive

```
E n1 n2 n3 n4
```

définit 5 champs :

```
A est défini des colonnes 1 à n1 -1
B de n1 à n2 -1
C de n2 à n3 -1
D de n3 à n4 -1
```

E de n4 à 80

On peut définir ainsi un maximum de 25 champs à l'aide des 24 numéros de colonne maximum de la directive E. Le dernier de ces champs est alors nommé "Y".

6.5.5 DIRECTIVE E? (VISUALISATION DES CHAMPS)

But :

Présenter les définitions de champ établies par la précédente directive E.

Syntaxe :

E?

Réponse : impression de 2 lignes

- numéros des colonnes limites de champ
- noms des champs définis présentés sous les numéros de colonne.

Exemple :

```
E?
12 20 24 77
A B C D E
A est défini de 1 à 11
B " 12 à 19
C " 20 à 23
D " 24 à 76
E " 77 à 80
```

6.5.6 CREATION DE LIGNE PAR CHAMPS

Après définition de champs, les directives I et R sous leurs formes : immédiate et entrée en mode Input, permettent d'entrer les caractères fournis en se limitant à un seul champ. Pour cela, on fait suivre le nom de la directive d'une lettre, nommant un champ défini au préalable.

Syntaxe :

```
Ich ligne
Ich
Rch ligne
Rch
```

ch : nom de champ : lettre de A à Y. Il ne doit pas y avoir d'espace " " entre le nom de la directive et le nom du champ, mais un espace ou un / est nécessaire avant la ligne.

Fonctionnement :

Pour chaque ligne normalement entrée, une ligne nouvelle est créée. Elle contient les caractères fournis, cadrés dans les limites du champ nommé. Si le nombre de caractères est supérieur à la longueur du champ, les caractères en excédent ne sont pas pris en compte.

Message :

NO FIELD Aucune définition préalable de champ par directive E. Ou définition annulée

UNDEFINED Champ non défini par la dernière directive E

TRUNCATED Trop de caractères fournis, il y a perte de caractères

Exemple :

```
E?
    5 10 20
A   B   C   D
IC STOP      STOP occupe les positions 10 à 13
IB
INPUT :
PUIS          PUIS les positions 5 à 8
CONTINUEZ    CONTI les positions 5 à 9
TRUNCATED    (NUEZ est perdu)
....
```

6.6 CREATION DE TEXTE PAR ZONE

Afin de simplifier l'entrée de caractères pour des textes n'acceptant des informations qu'entre certaines limites, il est possible de limiter la création de lignes à une zone définie. Cette méthode permet de rendre automatique la création d'espaces, sans risque d'erreur d'alignement et en protégeant les parties hors-zone de la ligne. Cette utilisation est possible après une définition de zone réalisée à l'aide de la directive Z (Voir chapitre ZONES).

Après une définition de zones les directives I et R, sous leurs deux formes, peuvent être suivis de la lettre Z pour limiter l'entrée de caractères à cette seule zone.

Syntaxe :

IZ ligne

IZ

RZ ligne

RZ

Fonctionnement :

Pour chaque ligne entrée, une ligne nouvelle est créée. Elle contient les caractères fournis, cadrés dans les limites de la zone. Si le nombre de caractères est supérieur à la longueur de la zone, les caractères en excédent ne sont pas pris en compte.

Messages :

NO ZONE Pas de zone définie

TRUNCATED Trop de caractères fournis : il y a perte de caractères.

Exemple :

Z?

2 72

IZ

INPUT :

PROTECTION COLONNE 1 ET FIN LIGNE

La ligne créée commence en colonne 2.

7 ACCES AUX FICHIERS ET UNITES E X T E R N E S

Le transfert de simples lignes ou de textes entiers pendant la session EDIT16 depuis ou vers des fichiers externes est réalisé par les directives G (GET) et W (WRITE).

La lecture de lignes depuis une unité ou un fichier associé à l'unité SI est réalisée par les directives I et R en mode lecteur

L'écriture de lignes d'impression vers une unité ou un fichier associé à l'unité LO est réalisée par la directive P en mode imprimante.

7.1 DIRECTIVE G (GET)

But :

Insertion derrière la ligne courante de lignes provenant d'un fichier externe.

Syntaxe :

G nomfic [-pw] [Ui|,Di|,Ei] [,NS] n1 n2

nomfic : nom du fichier externe (6 caractères max)

pw : nom de catalogue (2 caractères max)

Ui,Di,Ei : Unité symbolique ou Unité fonctionnelle disque qui contient le fichier. Par défaut l'unité fonctionnelle standard (ou celle définie dans la commande JOB de BOS-D).

NS : Non simultané. Ce paramètre est nécessaire si le fichier a été créé avec ce type de protection

n1 : numéro dans le fichier cité, de la lère ligne à insérer

n2 : numéro de la dernière ligne à insérer avec

1 < ou = n1 < ou = n2

Remarque :

La syntaxe concernant le fichier est la même que pour la commande EDIT et les directives W (WRITE) et M (MACRO)

Accès aux ARTICLES de FICHIER INDEXE :

La syntaxe de la directive permet la forme "nomartic.nomfic" en lieu et place de la forme "nomfic". Voir le chapitre ACCES AUX ARTICLES.

Fonctionnement :

Les lignes sont recherchées dans le fichier cité et insérées entre la ligne courante et la ligne suivante.
Le pointeur de ligne courante est positionné sur la dernière ligne insérée.

Si la valeur de n1 est supérieure au numéro de la deuxième ligne du fichier, la directive est ineffective.
Si la valeur de n2 est supérieure à ce numéro, l'insertion est réalisée jusqu'à cette ligne.

Message :

- ?
- le fichier n'a pas été trouvé
- Erreur de syntaxe sur n1 ou n2

Exemple :

```
G MISAJO-UR 3 35 - 33 lignes insérées
G AUTRE 13 13 - 1 ligne insérée
```

7.2 DIRECTIVE W (WRITE)

But :

Ecriture de lignes du texte en cours d'édition dans un fichier externe. Il y a création d'un fichier nouveau ou sur option : utilisation d'un fichier existant

Syntaxe :

```
W nomfic [-pw] [,Ui|Di|Ei] [,NS] (n|*) [R|E] [F80|Fnf]
```

Nomfic	nom, unité, type du fichier à créer ou utiliser. (voir syntaxe commande Edit ou directive G)
n	nombre de lignes, y compris la ligne courante
*	toutes les lignes du texte
R	Ecriture sur fichier existant avec Remplacement
E	Ecriture sur fichier existant avec Extension
F80 Fnf	Enregistrements de longueur fixe (voir chapitre particularités de fonctionnement)

Remarque : Accès aux ARTICLES de FICHIER INDEXE

La syntaxe de la directive permet la forme "nomartic.nomfic" en lieu et place de la forme "nomfic". Les options R, E et F ne sont pas alors accessibles. Voir le chapitre ACCES AUX ARTICLES de FICHIER INDEXE.

Fonctionnement :

Après création du fichier (ou ouverture de l'ancien fichier si option R ou E) les lignes sont transférées. Dans le cas d'écriture d'un nombre n de lignes la dernière ligne transmise devient ligne courante. Autrement, la ligne courante ne change pas. Dans le cas d'utilisation d'un fichier déjà existant, un paramètre R ou E devra être précisé, pour que la directive soit acceptée. Les lignes sont écrites dans le fichier.

- . Si R : les lignes écrasent et remplacent les lignes actuelles du fichier
- . Si E : les lignes sont placées à la fin du fichier sans destruction des lignes actuelles.

Message :

?

Un fichier avec ce nom existe déjà (sans paramètre R ou E : il s'agit d'une création et celle-ci n'est pas possible).

EOF

Fin de fichier (W avec nombre n de lignes supérieur au nombre restant. Le pointeur est positionné sur la fin du texte).

Exemple :

W TT1 15	Le fichier TT1 est créé ; il a 15 lignes.
W OLD 10 E	Dans le fichier OLD 10 lignes sont ajoutées à la fin.
W ANCIEN-NE 1000 R	les lignes sont écrites dans le fichier à la place des lignes existantes.

7.3 IMPRESSION DE LIGNES SUR L'UNITE D'IMPRESSION

7.3.1 DIRECTIVE P (PRINT) : MODE IMPRIMANTE

But :

Impression de lignes sur l'unité associée à l'unité symbolique d'impression

Syntaxe :

P: [n|*] [P] [T] [N]

n : nombre de lignes à imprimer y compris la ligne courante. Par défaut la ligne courante

* : impression de toutes les lignes du texte

P : Saut de page avant impression

T : Impression en titre du nom du fichier en cours d'édition

N : Sans numéro de ligne

Fonctionnement :

Sur l'unité ou le fichier associé à l'unité symbolique LO, les lignes sont imprimées ; chacune d'entre elles est précédée de son numéro de ligne. Un saut de page est réalisé toutes les 55 lignes.
Dans le cas d'impression d'un nombre n de lignes, la dernière ligne imprimée devient ligne courante.

Remarque :

Les options *, P, T et N ne sont pas disponibles avec la directive P (Print) normale.

Exemple :

```
* LO LP
...
P: 10 N Impression de 10 lignes sans numéros
P: 23 PT Saut de page, impression du nom du fichier et des 23
      lignes avec numéros
```

7.4 CREATION DE LIGNES DEPUIS UNE UNITE DE LECTURE CARTES

7.4.1 DIRECTIVES I (INPUT) ET R (REPLACE) EN MODE LECTEUR

But :

Lecture de lignes sur une l'unité associée à l'unité symbolique d'entrée SI

Syntaxe :

I: [n]

R: [n]

n: nombre de lignes à lire. Par défaut : lecture jusqu'à la rencontre d'une ligne vide.

Fonctionnement :

Les lignes lues sur l'unité symbolique SI, qui aura été associée par l'utilisateur à une unité de lecture de carte ou à un fichier, sont insérées après la ligne courante, (à la place, si directive R).

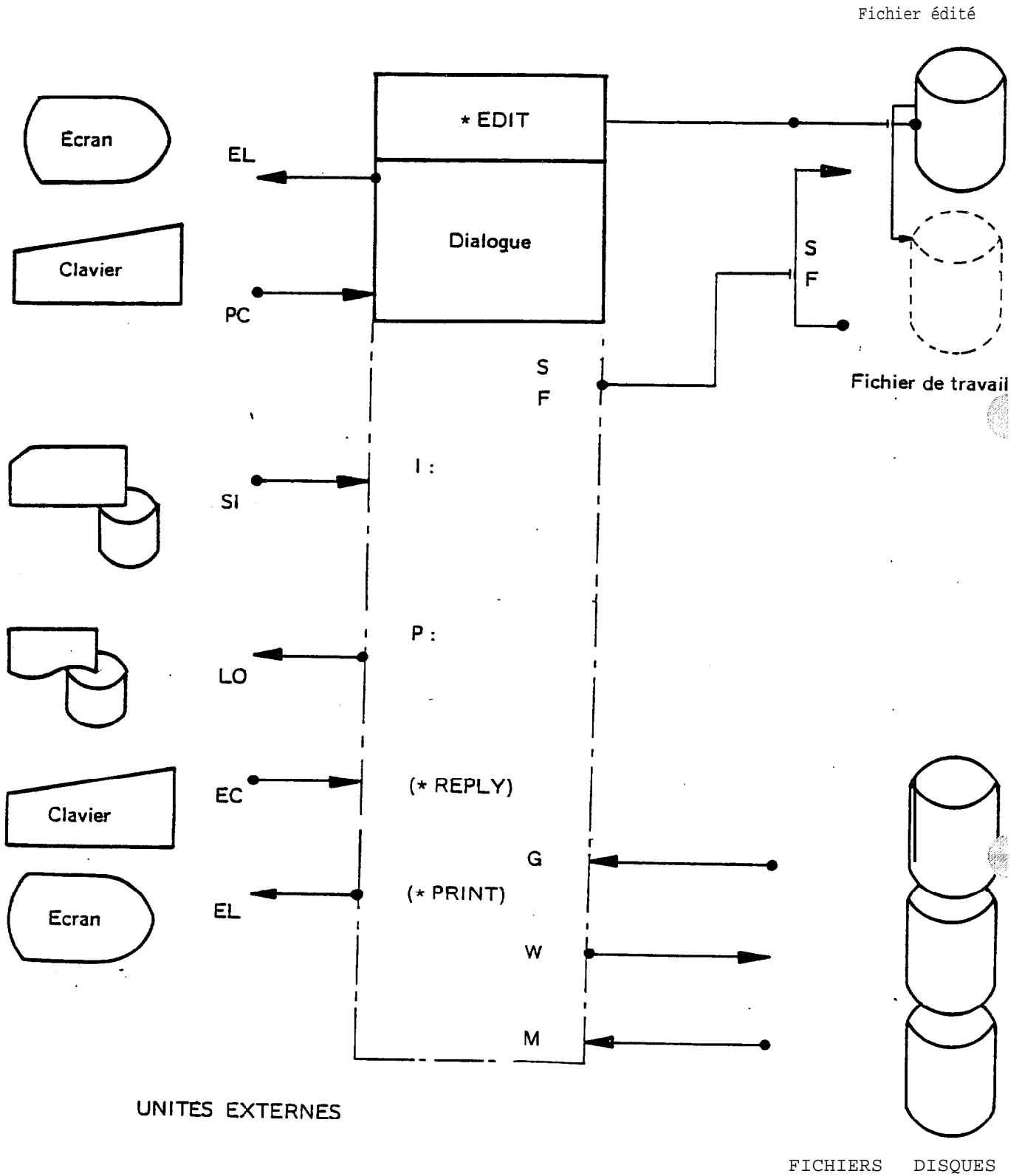
La fin de l'entrée de lignes sera provoquée par

- lecture d'une ligne vide : "CR" en col 1.
- lecture de n lignes
- lecture de code "EOT"
- appel opérateur ou défaut périphérique.

Exemple :

```
*SI CR  
.....  
i: 10 Lecture de 10 cartes sur Le lecteur.
```

7.5 SCHEMA UNITES SYMBOLIQUES ET FICHIERS



8 DEFINITION ET UTILISATION DE ZONE

Une ZONE définit une portion de ligne précisée par 2 numéros indiquant les positions de début et de fin de cette zone (numéros de colonne).
Après définition d'une zone par la directive Z, la portée des directives s'appliquant au contenu des lignes peut être limitée à cette zone.
On peut ainsi sélectionner certaines positions de la ligne sans considérer les autres, avec, comme conséquence, une protection de la partie hors-zone.

8.1 DEFINITION DE ZONE : DIRECTIVE Z

Cette directive peut prendre plusieurs formes

8.1.1 DEFINITION PROPREMENT DITE

But :

Définir une zone utilisable par les directives durant cette session.

Syntaxe :

Z np nd [I]

- np numéro première colonne
- nd numéro dernière colonne
avec $1 \leq np \leq nd \leq 80$
- I : zonage implicite.

Fonctionnement :

Une zone est définie, limitée par les numéros des positions np et nd, comprises dans la zone

- Si une zone était déjà définie, sa définition est annulée.
- Le paramètre I permet de définir une utilisation IMPLICITE de la zone : les futures directives étant automatiquement exécutées avec zonage, même si la lettre Z n'y est pas précisée.

Message :

? erreur de syntaxe

Syntaxe :

Z

Fonction :

Retour à l'état initial : pas de zone

8.1.3 VISUALISATION DE ZONE,

Syntaxe :

Z?

Fonctionnement :

- Impression des numéros des positions np et nd définissant la zone. - (ces numéros sont disponibles pour les macro directives dans ?PO)

Réponse :

Les 2 numéros

Message :

NO ZONE : Il n'y a pas de zone définie

8.1.4 EQUIVALENCE ZONE CHAMP

Syntaxe :

Z nomchamp [I]

Nomchamp : lettre nommant un champ défini
I : implicite (voir définition)

Fonctionnement :

- Cette forme, correspondant à une définition de zone, est utilisée pour définir une zone cadrée sur un champ préalablement défini par une directive E (Espacement). Voir CREATION DE TEXTE : Définition de champs. Comme toute définition de zone, cette forme annule et remplace la précédente définition de zone.

Message :

```
?           Erreur de syntaxe
NO FIELD    Champ inconnu
```

8.2 UTILISATION DE ZONE PAR DIRECTIVES

Toutes les directives concernées par le contenu de la ligne peuvent être exécutées en ne considérant que la portion de ligne comprise dans la zone définie par la directive Z.

Pour obtenir cet effet, le nom de la directive sera généralement suivi de la lettre Z, sans espace.

La limitation de l'effet de la directive aux seules positions incluses dans la zone est réalisée...

- si la lettre Z est indiquée explicitement à la suite du nom de directive
- systématiquement, même en l'absence de lettre Z, si la zone a été définie par la directive Z avec le paramètre I. (Implicit).

Le fonctionnement des directives concernées par la notion de zone est modifiée, si une zone est définie, au préalable :

D'une façon générale

- les chaînes sont recherchées à l'intérieur des limites de la zone
- les numéros de colonne sont considérés comme relatifs au début de la zone
- en l'absence de zone le message NO ZONE est émis.

Pour chaque directive concernée est donné ci-dessous la syntaxe simplifiée et le comportement en cas de définition de zone.

8.2.1 Directive C

```
CZ /chaîne 1/chaîne 2/n1 n2
```

La recherche de la chaîne est limitée à la zone définie. Le remplacement par la 2e chaîne est limité aux bornes de la zone. Le contenu de la ligne hors de la zone est inchangé.

8.2.2 Directives N et U

```
NZ /chaîne/
```

```
UZ /chaîne/
```

La recherche de la chaîne est limitée à la zone. (de même, si plusieurs chaînes sont utilisées).

IZ Ligne

IZ

Une ligne nouvelle est créée ; elle contient seulement les caractères fournis. Ils sont cadrés dans les limites de la zone. En mode "Input" chaque ligne est créée selon les mêmes conventions.

Message :

TRUNCATED les caractères entrés dépassent les limites de la zone.

Remarque :

Voir le chapitre CREATION DE TEXTE.

8.2.4 Directive R

RZ ligne

RZ

Le fonctionnement est identique à IZ, après destruction de la ligne courante.

8.2.5 Directive K

Seule la forme avec chaîne peut être utilisée avec zone.

KZ /chaîne/

La chaîne est recherchée à l'intérieur des limites de la zone. Les numéros de colonne rendus sont relatifs au début de la zone

8.2.6 Directive X

XZ n [l][nl|*]

Le numéro de colonne n est relatif au début de zone. La chaîne fournie est limitée à la fin de zone

YZ [n]/chaîne[/nl*]

Le numéro de colonne est relatif au début de zone. La chaîne est insérée dans les limites de la zone. Le décalage des caractères peut entraîner une perte en fin de zone. Le contenu de la ligne hors-zone n'est pas affecté.

8.2.8 Directive P

PZ [n]

Le seul contenu de la zone est imprimé en conservant sa position dans la ligne

8.2.9 Directive W

WZ nomfic n

Le fichier créé contient des lignes qui contiennent, cadrés à gauche, les caractères issus de la zone

8.2.10 Directive G

GZ nomfic np nd

Dans les lignes du fichier, seuls les premiers caractères lus sont considérés et, seulement jusqu'à concurrence de la longueur de la zone. Les lignes créées ne contiennent que ces caractères, cadrés dans la zone.

Remarque :

Pour chacune des directives ci-dessus, la lettre Z a été précisée. Le même effet sera obtenu, en omettant la lettre Z, si la zone a été définie par une directive Z comportant le paramètre I.

. Un message "IMPLICIT Z" est alors émis pour informer l'utilisateur d'une utilisation implicite de zone

9 MACRO-DIRECTIVES

9.1 GENERALITES :

L'utilisation de macro-directives doit être envisagée quand on veut par un appel unique entraîner l'exécution d'une suite de directives. Cette suite peut être exécutée

- . Soit séquentiellement
- . Soit avec un enchaînement et une logique déterminées par :
 - les réponses fournies à des questions
 - le résultat de l'exécution de directives d'édition
 - la valeur de paramètres d'appel

Après la programmation d'une macro-directive, l'utilisateur pourra ultérieurement en demander l'exécution en fournissant des valeurs pour paramétrer cette exécution.

- une macro-directive simple peut entraîner l'exécution d'une seule directive d'édition.

Une macro-directive complexe peut amorcer un dialogue personnalisé, au travers duquel seront réalisés des fonctions d'édition très élaborées par l'intermédiaire de directives qui seraient jugées fastidieuses à utiliser.

Une macro-définition est une suite de lignes constituant le corps de la macro-directive. Elles représentent :

- des chaînes de caractères qui seront ultérieurement traitées comme des directives
- des instructions de macro-définition
- des appels de macro-directives.

L'appel d'une macro-directive constitue une demande de génération (ou d'expansion) du contenu d'une macro-définition de même nom. Pendant cette génération l'exécution des instructions de macro-définition permet de définir des variables, de calculer des expressions, de réaliser des débranchements, de dialoguer avec l'utilisateur.

9.1.1 Programmation de MACRO-DIRECTIVES

La définition d'une macro-directive consiste à associer une suite de lignes appelées "corps" de la macro à une instruction de définition d'en tête.

Une macro-définition a la forme suivante :

```
* DEFen-tête  
  
Corps  
  
* ENDEF
```

L'en-tête contient la structure qu'il faudra respecter lors de l'appel. Cet appel utilisera simplement un nom ou associera des caractères à des paramètres formels, qui seront référencés dans le corps de la macro et remplacés au moment de l'expansion par des paramètres effectifs. L'instruction *DEF réalise la mémorisation de l'en-tête, c'est à dire de la forme possible de l'appel et du corps de la macro limité par l'instruction *ENDEF

9.1.2 Utilisation de MACRO-DIRECTIVES

L'appel sera réalisé ultérieurement par utilisation de cet en-tête derrière la directive % qui constitue l'appel de macro, éventuellement associé à des paramètres de substitution.

Exemples :

```
1      Après la définition  
          *DEFLISTE  
          P 23  
          *ENDEF
```

L' appel : %LISTE entraînera l'exécution de la directive P 23, unique directive du corps de la macro

```
2      Pour rechercher une chaîne dans un texte et imprimer son numéro de  
ligne, on peut créer la définition  
          *DEFLIGNE?  
          N. ?P1  
          L  
          *ENDEF
```

L'appel % LIGNE/ARTICLE/ provoquera la substitution de la chaîne
"/ARTICLE/" à la place du symbole ?P1 - l'appel se traduira par
l'exécution des directives
 N. /ARTICLE/
 L

9.2.1 INSTRUCTION DE DEFINITION

Syntaxe :

*DEFen-tête

L'en-tête est la chaîne de caractères apparaissant à la suite du nom de l'instruction.

Il peut comporter jusque 9 identificateurs de paramètres représentés par le caractère "?" et des caractères imprimables, qui permettront lors de l'appel, l'identification de la macro. L'ordre dans lequel ces éléments sont placés est essentiel puisque c'est uniquement leurs positions qui permettent à l'exécution de les distinguer. Ils seront utilisés ultérieurement sous la forme. ?Pn avec n évoluant de 1 à 9.

Exemples.

```
*DEF?V?  
*DEFLIRE? , ?
```

peuvent être appelés respectivement par

```
%1V2 ou %60V79
```

```
%LIRE 9,350
```

Remarque :

Une macro-définition peut ne comporter aucun "?" afin qu'à la génération aucun paramètre ne soit fourni. Dans ce cas, les séquences générées lors des différents appels peuvent être déterminés, le cas échéant, par d'autres éléments tels que

- le résultat d'une directive d'édition
- la réponse à une instruction de dialogue.

Remarque :

L'utilisateur familiarisé avec la programmation de directives complexes, trouvera en fin de chapitre les règles utilisées par EDIT16 pour classer les macros définitions.

Redéfinition :

Une macro avec un même en-tête qu'une macro déjà définie remplace la définition précédente et l'annule donc.

9.2.2 INSTRUCTIONS DE FIN DE DEFINITION

Syntaxe :

*ENDEF

Cette directive finit la définition ouverte par *DEF.

Il est alors possible de passer à une autre définition ou de revenir en environnement EDIT.

Les éléments compris entre l'en-tête *DEF et la fin de définition *ENDEF sont des lignes qui peuvent être considérées comme

- des instructions de macro-directives si la ligne commence par le caractère "*"
- sinon, des lignes à mémoriser telles quelles. A l'appel de la macro, elles seront considérées comme des lignes à destination d'EDIT16 ; directives ou texte.
- La directive % d'appel de macro-directive peut apparaître dans le corps d'une macro-directive. L'appel récursif à la macro en cours est possible.
- des lignes à générer si elles commencent par le caractère " " ("antislash"), ces lignes seront générées après suppression du " " sans être interprétées.

Ainsi la ligne

```
*ENDEF
```

donne lieu à la génération de la ligne

```
*ENDEF
```

et n'est pas reconnue comme une fin de macro-définition.

9.2.4 INSTRUCTIONS DE MACRO-DIRECTIVES

Elles sont identifiées par le caractère "*". Elles peuvent se classer en 3 types

- Affectation de variables :

```
*V
```

- Rupture de séquence

```
*SKIP  
*IF SKIP  
*KILL
```

- Dialogue

```
*PRINT  
*LPRINT  
*T  
*REPLY
```

Elles sont décrites au cours des paragraphes suivants.

9.2.5 UTILISATION DES PARAMETRES

A l'intérieur du corps de la macro, une ligne quelconque peut faire référence aux paramètres. En nombre maximum de 9, et numérotés P1 à P9 dans leur ordre d'apparition.

La forme P0 suit les mêmes règles d'utilisation. Elle sera étudiée plus loin.

?Pn est le remplacement du paramètre formel par le paramètre effectif de rang n

?Ln est le remplacement par la longueur du paramètre exprimée en

nombre de caractères

?Tn est le remplacement par le type du paramètre effectif de rang n

Les différents types considérés sont :

Type 1 : nombre hexadécimal : chiffres de 0 à F précédés de "".

Type 2 : nombre décimal précédé ou non d'un signe + ou -

Type 3 : chaîne de caractère : caractères encadrés par des doubles apostrophes

Type 4 : symbole : chaîne de caractères alphabétiques et numériques, le 1er alphabétique.

Type 5 : autre

Remarque : le paramètre vide a comme longueur 0 et comme type 5.

9.2.6 DEFINITION DE VARIABLES : INSTRUCTION *V

Une fois définies dans le corps d'une macro-directive, les variables sont utilisables dans toute autre macro.

Les variables sont identifiées par un nom de la forme Vnln2 dans la plage de V01 à V99.

. La variable V00 réservée à un usage spécial ne doit pas être définie : voir plus loin.

. L'instruction de définition *V permet d'attribuer à une variable un type et une valeur.

- Type ENTIER si définie par

*Vnln2 = expression arithmétique

La valeur attribuée à la variable est le résultat du calcul de l'expression arithmétique entière.

Exemple :

```
*V01 = 45    donne 45
*V02 = -33/4    -8
```

- Type CHAINE si définie par

*Vnln2 EQ expression chaîne

Le nombre maximum de caractères est 78

Exemple

```
*V03 EQ CHAINE
*V04 EQ '124
```

- Une même variable peut être définie plusieurs fois. Chaque redéfinition annule la définition précédente, en valeur et en type.

```
*V05 = 15    entière
*V05 EQ DOUZE chaîne
```

UTILISATION DES VARIABLES

La référence à une variable se fait par la syntaxe :

@Vnln2

qui représente la valeur actuelle de la variable.

Exemple :

*V06 = 20

P@V06 donne P20

9.2.8 EXPRESSIONS ARITHMETIQUES

Une expression arithmétique est un assemblage de variables entières et de nombres décimaux signés reliés entre eux par des opérateurs arithmétiques.

Les quatre opérateurs arithmétiques :

+ - * /

ont même priorité et l'ordre d'exécution des opérations est de la gauche vers la droite.

Le caractère "*" apparaissant dans une expression indique la multiplication des éléments situés à sa gauche (multiplicande) par l'élément situé à sa droite (multiplicateur).

Exemple :

15 + 2 * - 3 + 47

Multiplicande : 15 + 2, multiplicateur -3, résultat : -4.

Le caractère "/" apparaissant dans une expression indique la division entière des éléments situés à sa gauche (dividende) par l'élément situé à sa droite (diviseur).

Exemple :

16378 / -2 + -4

Dividende : 16378, diviseur : -2, résultat : -8193

La valeur d'une expression arithmétique est un nombre décimal signé calculé au moyen de l'expression elle-même.

Exemple :

*V06 = @V05 + 12 Valeur résultat :27 si V05 avait la valeur 15.

9.2.9 EXPRESSIONS CHAINE

Toute chaîne de caractères, assemblage de variable chaînes et caractères est une expression chaîne.

Une expression arithmétique peut être considérée comme une expression chaîne ; la réciproque n'étant évidemment pas vraie.

La valeur d'une expression chaîne est la chaîne elle-même.

Exemple :

*V06 EQ @V03 @V05 donne CHAINE DOUZE

9.2.10 RUPTURE DE SEQUENCE

9.2.10.1 INSTRUCTION *SKIP

Syntaxe :

*SKIP expression arithmétique

Instruction de rupture inconditionnelle, elle interrompt la séquence du traitement.

Le résultat du calcul de l'expression ou la simple valeur n fournie est un déplacement relatif à la ligne traitée.

Exemple : ainsi La séquence

```
*SKIP 3
....   Ligne ignorée.
....   Ligne ignorée.
....   Reprise du traitement
```

provoque un saut à la 3e ligne suivant l'instruction *SKIP.

Une valeur négative provoque un saut en arrière.

Une valeur aboutissant hors du corps d'une macro provoque une erreur (voir messages en annexe).

La valeur 0 provoque la sortie de la macro ; le traitement reprend au point d'où a été appelée la macro.

9.2.10.2 INSTRUCTION *IF SKIP

Syntaxe :

*IF relation SKIP expression arithmétique

Instruction de rupture conditionnelle, elle poursuit le traitement, en séquence ou à une ligne indiquée suivant qu'une relation est vraie ou fausse.

Lorsque la relation est vérifiée, *IF se résume à un *SKIP ; dans le cas contraire, le traitement continue à la ligne suivante.

Deux types de relations peuvent intervenir. Elles conduisent à deux formes.

9.2.10.3 Relations arithmétiques

Trois opérateurs de relation sont utilisés

<	inférieur
=	égal
>	supérieur.

Une relation est alors formée de l'assemblage

expression arithmétique < ou = ou > expression arithmétique

Exemple :

```
*IF ?P1 = 4 SKIP 3  
*IF ?L2 = 0 SKIP -12
```

9.2.10.4 Relations logiques

Deux opérateurs sont utilisés

EQ pour équivalent
NE pour non équivalent

Une relation est alors formée de l'assemblage

expression chaîne EQ ou NE expression chaîne

Les 2 expressions sont comparées entre elles, caractère par caractère.
Deux chaînes sont équivalentes si elles ont la même longueur et sont composées de la même séquence de caractère.

Exemple :

```
*IF ?P2 EQ A SKIP 0  
*IF @V10 EQ @V09 SKIP 5
```

Note :

La relation suivante est fausse : 3 + 4 EQ 7

9.2.10.5 INSTRUCTION *KILL

Syntaxe :

*KILL expression chaîne

- Abandon de la génération de la macro-directive
- Impression de l'expression chaîne, qui peut représenter un message d'erreur
- Retour à l'environnement édition, signalé par le message EDIT :

Note :

La différence avec un *SKIP 0 réside dans le fait que ce dernier ne fait pas abandonner la génération, mais revient à l'instruction qui suit l'appel.

9.2.11 INSTRUCTIONS DE DIALOGUE

9.2.11.1 INSTRUCTION *PRINT

Syntaxe :

*PRINT texte

But :

Impression sur l'unité affectée à l'unité symbolique EL du texte précisé. Celui-ci est suivi des caractères "RC" et "LF" qui positionnent le curseur en début de ligne suivante. La chaîne de caractères qui constitue le texte ne doit pas dépasser 71 caractères.

9.2.11.2 INSTRUCTION *LPRINT

Syntaxe :

*LPRINT texte

But :

Comme *PRINT, impression de texte sur EL mais sans que soient ajoutés les caractères de positionnement du curseur. Le curseur reste positionné en fin de texte.

9.2.11.3 INSTRUCTION *REPLY

Syntaxe :

*REPLY [n]

But :

Lecture de caractères sur l'unité associée à l'unité symbolique EC. Ces caractères sont disponibles dans ?P0. (Voir paragraphe suivant)

- En l'absence de paramètres, une ligne entière est lue, jusqu'à un retour chariot provoqué par la touche "Return" du clavier.
- Si le nombre de caractères "n" est précisé, la lecture s'arrête quand ce nombre est atteint, quels que soient les caractères entrés. Le curseur reste positionné sur le caractère suivant le dernier lu.

Remarque :

L'enchaînement de *LPRINT et de *REPLY n permet de rester sur une même ligne.

3.2.11.4 INSTRUCTION *T

Syntaxe :

*T liste de numéros séparés par des blancs

But :

Définition d'une tabulation utilisable lors de l'impression d'une ligne par *PRINT ou *LPRINT.
Le caractère de tabulation est ">". Si on veut imprimer le caractère ">", il suffit de le doubler. ">>"



Après exécution d'une directive d'édition depuis le corps d'une macro, il est possible de

- savoir si cette directive s'est exécutée normalement
- traiter la réponse éventuelle que cette directive imprime normalement sur la console de dialogue.

9.2.12.1 VARIABLE V00

La variable V00, réservée à cet usage, est rendue avec la valeur.

1 si la directive s'est déroulée normalement
0 si l'exécution de la directive a entraîné une condition particulière

Ces conditions particulières sont celles qui correspondent à la production de messages d'information ou d'anomalies tels que "TOP" "NO CHANGE" "?" etc... (voir liste en annexe), à l'exception du message EDIT. Elles ne sont pas dépendantes du fait que le mode normal ou le mode sans message ait été demandé pour la directive (caractère "," ou ";").

Exemple :

```
N 20
*IF @V00 = 0 SKIP 0
```

On termine la macro en cours si le N 20 a fait rencontrer la fin du texte : message EOF.

9.2.12.2 PARAMETRE P0

Ce paramètre, spécialisé et distinct des paramètres normaux, contient le texte sanctionnant normalement la fin de l'exécution d'une directive d'édition ou de l'instruction de dialogue *REPLY. Le texte est disponible dans P0, même si la directive a été demandée avec le mode muet (caractère "." ou ";"). Le texte obtenu est :

- en général : la ligne courante
- pour les directives L et 0 : un numéro de ligne.
- pour les directives K, E? Z? : un ou plusieurs numéros de colonne séparés par des espaces
- pour la directive X : la chaîne demandée.
- pour l'instruction *REPLY : la ligne entrée par l'utilisateur

Comme pour les paramètres d'appel, le contenu de la chaîne peut être utilisé : ?P0, aussi bien que sa longueur ?L0 et son type ?T0

9.2.13 EXEMPLE DE MACRO-DIRECTIVE

Macro-directive pour supprimer des lignes jusqu'à la ligne contenant une chaîne spécifiée.

Son utilisation se fera par l'appel

```
%D /CHAINE/
```

Sa définition peut être la suivante.

```
*DEFD ? . Le paramètre est la chaîne avec //
L.
*V01 = ?P0 . V01 : numéro de ligne courante
*IF @V01 > 0 SKIP2 . Si on est sur ligne 0 . . .
*V01 =@V01 + 1 . . . Aller à la ligne suivante
N. ?P1 . Recherche la chaîne
*IF @V00 =0 SKIP 0 . Si pas trouvée, sortie de macro
L.
*V01 = ?P0-@V01 . Calcul du nombre de ligne à . . .
U. @V01 . . . remonter . . .
D @V01 . . . et supprimer
P . Impression nouvelle ligne courante
*ENDEF
```

9.2.14 CAS DE LA DIRECTIVE A (AGAIN)

Lorsque la directive A suit l'appel d'une macro-directive, elle répète effectivement celle-ci.

```
%L /36/
A 60
```

Lorsque la directive A est située dans le corps d'une macro-directive, elle ne répète que la dernière directive générée. Ainsi

```
*DEFL?
N.?P1
C. /?P1//
A3
*ENDEF répète 3 fois la seule directive C.
```

9.2.15 UTILISATION DE CARACTERES NON-IMPRIMABLES

Seuls les caractères imprimables peuvent normalement apparaître dans le corps d'une macro-directive.

La fonction suivante permet de définir un caractère non imprimable.

Syntaxe :

```
?Hxx
```

x : chiffre hexadécimal : de 0 à F

But :

Insertion d'un caractère dont le code ASCII est 'xx', en place de la référence ?Hxx

Exemples 1

```
*DEFINCODE8F?  
Y. ?P1 /?H8F/  
*ENDEF
```

La directive "%INSCODE8F 12" réalise l'insertion du caractère de code hexadécimal '8F' en colonne 12.

Exemples 2

```
*DEFTITRE  
*PRINT ?HOC . . . . . TITRE . . . . .  
*ENDEF
```

Le caractère 'OC' réalise l'effacement de l'écran sur un écran VAP24. Sur la 1ère ligne de l'écran s'imprime en suite :

```
. . . . . TITRE . . . . .
```

9.3 CREATION ET APPEL DE MACRO-DIRECTIVES

Après programmation d'une macro-directive, sa prise en compte sera réalisée dans l'environnement MACRO-DEFINITION, son exécution dans l'environnement MACRO-GENERATION

9.3.1 DIRECTIVE M (MACRO)

But :

Passer dans l'environnement macro-définition. Définir comment seront fournies les macro-définitions.

Syntaxe :

1ère forme M

2ème forme M nomfic[-pw] [,Ui|Di|Ei] [NS]

Nom et caractéristiques d'un fichier existant (voir syntaxe commande EDIT)

Fonction :

1ère forme : M sans paramètre. Les macro-définitions sont créées directement à partir du clavier, après que soit apparu le message DEFMAC :
Quand, après une instruction *ENDEF, une ligne vide est entrée, il y a retour à l'environnement d'édition et émission du message EDIT.
Les définitions ne seront pas mémorisées au delà de la fin de session.

2ème forme : Les macro-définitions sont lues sur le fichier nommé. Elles auront été préalablement créées sur ce fichier, en général avec EDIT16.

Remarque : Macro-définition sur ARTICLE de FICHER INDEXE
La syntaxe de la directive M permet la forme "nomartic.nomfic" en lieu et place de la forme "nomfic". Voir le chapitre ACCES AUX ARTICLES.

Réponses :

? Le fichier n'a pas été trouvé

EDIT : Les macro-définitions ont été lues et prises en compte. (Tout le fichier a été traité).

DEFMAC : Fichier incomplet, l'utilisateur est placé dans le même cas que pour la 1ère forme

Messages :

Des messages de la forme ERMnn peuvent être émis, en cas d'erreur ; voir leur signification en annexe. IL y a lieu de corriger le fichier macro et de réémettre la directive M.

Exemple 1 :

```
Deux macro-définitions entrées en ligne
M
DEFMAC :
*DEFMA ? ?
Corps
*ENDEF
*DEFMB ?
Corps
*ENDEF
Ligne vide
EDIT :
```

Exemple 2 :

```
Définition de macro-directives contenues sur un fichier
M APPLI-MA
EDIT :
```

Le fichier APPLI-MA contient une ou plusieurs macros sous une forme identique à plus haut en respectant la séquence.
*DEF, ...corps, *ENDEF, *DEF. ...corps, *ENDEF etc.....

Remarque :

La taille maximale et le nombre de macro-directives que l'on peut définir est fonction de la taille de la zone mémoire disponible dans la partition où est implanté l'éditeur.

But :

Exécution d'une macro-directive, définie et créée.

Syntaxe :

%macro-directive

Fonction :

Recherche d'une définition correspondant à la suite de caractères fournie.
Substitution des paramètres effectifs. Génération de la macro-directive.
Présentation immédiate des lignes générées considérées comme des lignes directives ou texte pour EDIT16.

Messages :

Ils peuvent être de 3 types

- Des messages de la forme ERMnn au cas d'erreur au cours de l'expansion de la macro (voir leur signification en annexe). Ils peuvent être suivis du numéro de la ligne en erreur dans la macro en cours.
- Les erreurs sur les directives d'édition contenues dans le corps de la macro-définition. Ils entraînent l'abandon de la génération.
- Les messages d'information concernant les directives d'édition. Ils n'entraînent pas l'abandon de la génération.

9.4 COMPLEMENT : CLASSEMENT DES MACRO-DEFINITIONS

Afin de définir de façon unique l'en-tête correspondant à un appel de macro-directive lorsque plusieurs en-têtes peuvent être choisis ; les 5 règles suivantes sont utilisées dans l'analyse de l'en-tête.

1. La recherche s'effectue de gauche à droite.
2. Un séparateur (au moins) doit s'intercaler entre 2 "?". Le doublement de caractère "?" conduit à le considérer comme un caractère normal. Exemple "QUI???" est considéré comme : le séparateur "QUI?" puis l'indicateur "?"
3. Entre un indicateur de paramètre "?" et un séparateur, c'est le séparateur qui est choisi. - exemple : A = ? + ? est choisi face à ? = ? + ?.
4. En cas d'ambiguïté dans le choix des paramètres, le groupage à droite est effectué. Exemple : avec l'en-tête "AX?,?" l'appel "AXU,V,W" choisit comme paramètre ?P1 : "U" et comme paramètre ?P2 : "V,W".
5. Le nombre d'espaces d'une suite d'espace n'est pas significatif. Autrement dit : les espaces après le premier ne sont pas considérés lors de l'appel. Un espace au moins devra figurer. Les espaces à l'intérieur d'un paramètre effectif sont tous significatifs.

QUELQUES EXEMPLES DE MACRO-DIRECTIVES

1. LA MACRODIRECTIVE %N RECHERCHE LES OCCURENCES D'UNE CHAINE AVEC IMPRESSION DU NUMERO DE LIGNE DE CHAQUE OCCURENCE :

DEFINITION :

```
*DEFN ?  
T  
N./?P1/  
*IF @V00=0 SKIP 0  
L  
*SKIP 3  
* ENDEF
```

EXECUTION :

```
%N MOT  
1  
19  
EOF
```

LES LIGNES 1 ET 19 CONTIENNENT LA CHAINE MOT

2. LA MACRODIRECTIVE %C ANALYSE UN FICHER POUR REPERTORIER LES OCCURENCES D'UNE CHAINE ; LES NUMEROS DES LIGNES SONT RANGES DANS DES LIGNES CREEES EN FIN DE TEXTE ; CELLES-CI SE TERMINENT PAR =.

DEFINITION :

```
*DEFC ?  
B.  
I =  
T  
N. ?P1  
*IF @V00=0 SKIP 0  
O.  
*V01 = ?P0  
B.  
*IF ?L0 < 67 SKIP 2  
I =  
C. /=/@V01,=/  
J. @V01  
*SKIP 9  
*ENDEF
```

EXECUTION :

```
%C /PROCEDURE/
```

PAR EXEMPLE LES DERNIERES LIGNES CONTIENNENT :

```
12,23,56,132,184,222,239,250,272,289,312,=  
350,387,=
```

.3. LA MACRODIRECTIVE %P IMPRIME LE TEXTE JUSQU'A LA LIGNE CONTENANT
UNE CHAINE DONNEE

DEFINITION :

```
*DEFP ?  
L.  
*V01 = ?P0  
*IF @V01 > 0 SKIP 2  
*V01=@V01 + 1  
N. ?P1  
*IF @V00=0 SKIP 0  
L.  
*V01=?P0 @V01  
U.@V01  
*V01 = @V01 + 1  
P@V01  
*ENDEF
```

UTILISATION :

```
*P /PROCEDURE/
```

.4. LA MACRODIRECTIVE %XD (X DESTRUCTIF) REALISE LA FONCTION DE
LA DIRECTIVE X EN REMETTANT A BLANC LES CARACTERES PRESENTES.

DEFINITION :

```
*DEFXD ? ?  
*V01 = ?P1 + ?P2 - 1  
X ?P1 ?P2  
*V02 EQ ?P0  
Z ?P1 @V01  
CZ /@V02//  
Z  
*ENDEF
```

UTILISATION :

```
%XD 12 4
```

LES CARACTERES DES COLONNES 12,13,14,15 SONT TRANSFORMES EN BLANCS
ET IMPRIMES.

REMARQUE: UNE MACRO SEMBLABLE PERMETTRAIT DE SUPPRIMER LES CARACTERES
AU LIEU DE LES METTRE A BLANCS : REMPLACER "Z ?P1 @V01" PAR "Z ?P1 80"
ET SUPPRIMER LA 2EME LIGNE.

.5. LA MACRODIRECTIVE %SERIAL CREE UNE SERIALISATION DES LIGNE
D'UN TEXTE PAR INSERTION DE CARACTERES NUMERIQUES.

SYNTAXE :

```
*SERIAL COLONNE VALEUR PAS
```

UTILISATION

L'APPEL %SERIAL 75 0001 1 INSERE EN COLONNE 75 UNE
CHAINE DE 4 CARACTERES NUMERIQUES. A CHAQUE NOUVELLE LIGNE CETTE
VALEUR EST AUGMENTEE DE 1.

L'APPEL %SERIAL 2 010 5 INSERE SUCCESSIVEMENT LES
VALEURS 010,015,020, ETC... LES COLONNES 2 ET SUIVANTES SONT
DECALEES DE 3 POSITIONS.

DEFINITION :

```
*DEFSERIAL ? ? ?  
*V01 EQ ?P2  
*V02 EQ ?L2  
*V03 EQ ?P3  
Y. ?P1 /@V01/ INSERTION COLONNE DE LA VALEUR  
*V01 = @V01+@V03 ADDITION DU PAS  
%NUMSER @V01 APPEL MACRO NUMSER  
N;  
*IF @V00 = 0 SKIP 2  
SKIP =5  
*PRINT FIN DE TEXTE  
*ENDEF  
*DEFNUMSER ? INSERER DES ZEROS DEVANT LES CHIFFRES  
*IF ?L1=@V02 SKIP 0  
*V01 EQ 0?P1  
*NUMSER @V01 APPEL RECURSIF  
*ENDEF
```

REMARQUE :

IL FAUT NOTER QUE L'INTRODUCTION D'UNE SERIALISATION EN FIN DE
LIGNE PEUT ENTRAINER UNE AUGMENTATION CONSIDERABLE DE L'ENCOMBREMENT
D'UN TEXTE A CAUSE DU NOMBRE DE CARACTERES BLANCS INTRODUIITS.

.6. LA MACRODIRECTIVE %QMAC LISTE LES DEFINITIONS D'EN TETE
DE MACRO CONTENUES DANS UN FICHER DESIGNÉ.

DEFINITION :

```
*DEFQMAC
*LPRINT QUEL FICHER ?? :
*REPLY
*V01 EQ ?P0
L.
*V10 EQ ?P0
*V10 = @V10 + 1
G, @V01 1 10000
L.
*V11 = ?P0+@V10 + 1
T
N. @V10
Z 1 4
CZ,/*DEF/*DEF/ @V11 1
Z
T
N. @V10
D,@V11
*PRINT FIN
*ENDEF
```

EXECUTION :

```
%QMAC
QUEL FICHER ? :SMACMR
*DEFN ?
*DEFC ?
*DEFP ?
*DEFXD ? ?
*DEFSERIAL ? ? ?
*DEFNUMSER ?
*DEFQMAC
FIN
```

REMARQUES :

LES LIGNES DU FICHER DESIGNÉ SONT AMENÉES EN MÉMOIRE PUIS
DETRUITES.

IL EST UTILE DE PLACER "%QMAC" DANS CHAQUE FICHER DE
MACRODIRECTIVES.

10 PARTICULARITES DE FONCTIONNEMENT

10.1 AUTRES COMMANDES DE L'EDITEUR

En plus de la commande EDIT l'éditeur autorise l'utilisation de trois autres commandes.

10.1.1 COMMANDES GO ET STOP

But :

Ces commandes permettent de revenir dans l'environnement EDIT après qu'un défaut périphérique ou un appel opérateur (touche BREAK sur le clavier) ait rendu contrôle au dialogue système.

Syntaxe :

GO

Fonction :

Le contrôle est rendu à EDIT16 - l'exécution de la directive est reprise, le cas échéant, au point où elle a été interrompue.

Syntaxe :

STOP

Fonction :

Le contrôle est rendu à EDIT16. Celui-ci ne reprend pas le traitement interrompu et passe en environnement EDIT en attente d'une directive.

10.1.1.1 Utilisation de l'appel opérateur (BREAK)

On utilise cette touche dans les cas suivants :

- Interruption des échanges sur l'unité symbolique SI (mode lecteur) avec lecteur de cartes ou dérouleurs de bandes magnétiques.
- Interruption des éditions sur l'imprimante ou le périphérique de dialogue

- Interruption d'une boucle commandée par A (AGAIN)
- Passage au dialogue système ; par exemple pour réaliser une affectation d'unité symbolique SI ou LO

Il est NECESSAIRE de revenir à l'environnement EDIT par une commande GO à une commande STOP afin de terminer correctement le traitement interrompu. Le fichier intermédiaire serait en effet perdu, si une de ces commandes n'était pas émise, suivie d'une directive S ou F.

10.1.2 COMMANDE SPACE

But :

Définition d'un espace virtuel autre que l'espace virtuel standard prévu à la mise en oeuvre

Syntaxe :

SPACE i, U

i :	taille de l'espace virtuel demandé
	0 pour un texte de 1364 lignes
1	2729
2	5460
3	10921

U : Désigne l'unité disque qui supportera le fichier de travail : espace virtuel. Une unité symbolique dans la plage U1 à UF, une unité fonctionnelle dans la plage D2 à DF et E1 à EF

Exemple :

Création d'un espace virtuel pour un texte d'environ 10.000 lignes sur D8.

```
* U2 D8
* SPACE 3, U2
```

Réponse :

* ou > : la commande a été bien exécutée. Le contrôle est rendu au système qui attend une nouvelle commande : normalement EDIT

Message :

EREi Une erreur a été détectée voir sa signification en appendice.

Remarques :

- le fichier contenant l'espace virtuel standard, de nom <<VI<< n'est pas détruit.
- Le fichier créé pour l'espace virtuel spécial est nommé <<V2<<.

- Il est détruit en fin de session
- Le catalogue utilisé pour ces fichiers est
 - . "AB" avec BOSD
 - . Attribué par le système, selon l'utilisateur sous les systèmes multiconsoles. C'est ce nom qui apparaît sous la forme PUBW = n au moment de l'EDIT
 - l'unité fonctionnelle disque doit satisfaire aux conditions précisées au chapitre MISE EN OEUVRE
 - La commande SPACE est obligatoire s'il n'existe aucun espace virtuel standard.
 - L'indicateur i, précisant la taille de l'espace virtuel spéciale, ne doit pas être supérieur à la valeur définie lors de l'intégration de l'éditeur : voir MISE EN OEUVRE.

10.2 FICHIERS DE TRAVAIL DE L'EDITEUR

EDIT16 permet d'éditer des textes contenus dans des fichiers de type séquentiel ou dans des articles de fichier indexé. Il commence toute édition par le transfert du fichier ou de l'article à éditer, s'il existe dans un fichier de travail, dit "espace virtuel". Cela, afin d'effectuer avec des performances optimales les opérations d'insertion de suppression de lignes etc... Il est de type "direct" et est organisé en pages de 1024 mots. Le nombre des pages est fonction de la taille maximum du fichier à éditer. Il est défini selon les indications contenues dans les premiers mots de l'éditeur (voir MISE EN OEUVRE) et selon les indications éventuelles émises par une commande SPACE. Ce fichier a comme nom <<VI<<. Le nom de catalogue utilisé est "AB" avec le système BOS-D, ou un numéro d'utilisateur, attribué par le système avec les systèmes multiconsoles c'est ce numéro qui est présenté sous la forme PUBW = n au moment de la commande EDIT ou des directives S et F.

Pendant toute la durée de l'édition le fichier ou l'article source nommé dans la commande EDIT n'est pas modifié.

A la rencontre d'une directive F (FILE) OU S (SAVE) pour une session d'édition sur un fichier séquentiel, les opérations suivantes sont réalisées afin d'éviter la perte du fichier origine en cas d'incident.

- . Le fichier source nommé à l'EDIT, est renommé <<TE<<, avec le même nom de catalogue sous BOS-D, avec le nom défini par PUBW = sous système multiconsole.
- . Un fichier de même nom que le fichier source est alors créé, de type séquentiel.
- . Le fichier espace virtuel est transféré dans ce nouveau fichier
- . Si le transfert s'est terminé normalement, le fichier <<TE<< est détruit. Si un incident apparaît pendant cette copie, l'utilisateur pourra récupérer son fichier initial et recommencer l'édition, après correction de la cause d'erreur.

Exemple :

```
(Sur système multiconsole TSM)

F                               directive F de l'utilisateur
PUBW - 5                        imprimé par EDIT16 si TSF
NOMFIC-CT                       " " "
ERB 41 6021                      imprimé par le système
    - message indiquant la saturation de la FU disque utilisée

Emettre les commandes suivantes

CLOSE                            (EOJ sous BOSD)

SI NOMFIC-CT

DELETE SI

SI <<TE<<-5                      (<<TE<<-CT sous BOS-D)

RENAME SI,NOMFIC-CT

CLOSE SI

Puis réaliser le "nettoyage" de la FU utilisée.
```

Remarque :

Dans le cas d'édition d'ARTICLES de FICHER INDEXE, les opérations réalisées sont différentes. On se reportera au chapitre : ACCES AUX ARTICLES DE FICHER INDEXE.

10.3 FICHIERS A ENREGISTREMENTS DE LONGUEUR FIXE.

10.3.1 LONGUEUR 80 CARACTERES.

Quand cela est rendu nécessaire par une utilisation ultérieure particulière, il est possible de créer des fichiers composés d'enregistrements de longueur fixe, c'est à dire comprenant strictement les 80 caractères d'une image de carte au lieu d'être de longueur variable et terminés par les caractères RC et LF de fin de ligne.

Ces fichiers seront réutilisables par EDIT16.

La création de tels fichiers est réalisable par adjonction d'un paramètre en fin de la directive F ou de la directive W. Ce paramètre apparaîtra sous la forme "F80".

- La syntaxe de la directive F (FILE) devient

```
F [F80]
```

- La syntaxe (simplifiée) de la directive W (WRITE) devient

W nomfic n [F80]

Pour la forme complète du W (WRITE) voir chapitre : ACCES AUX FICHIERS EXTERNES.

10.3.2 LONGUEUR INFERIEURE A 80 CARACTERES

Avec la directive W (WRITE), des enregistrements d'une longueur inférieure à 80 caractères peuvent être demandés pour des cas très spéciaux. En effet le fichier créé, s'il est réutilisé à l'aide d'EDIT16, apparaît totalement désorganisé, car

- 1) seuls les *nf* premiers caractères de chaque ligne ont été transférés pendant l'écriture sur le nouveau fichier
- 2) à la lecture, ultérieurement par commande EDIT ou directive G (GET) le fichier est considéré comme comportant des enregistrements de 80 caractères, parce qu'il ne comporte pas de caractères RC et LF. Aucune information cependant n'est perdue à ce niveau.
Cette utilisation n'est pas compatible avec l'utilisation de l'option : Réécriture avec R ou E.

- la syntaxe (simplifiée) de la directive W (WRITE) est alors

W nomfic n [F*nf*]

F*nf* : enregistrements de *nf* caractères avec *nf*<80.

Pour la syntaxe complète de la directive W (WRITE) voir au chapitre : ACCES aux FICHIERS EXTERNES.

11 ACCES AUX ARTICLES DE FICHER INDEXE

- Le traitement d'articles de fichier indexé permet d'accéder à la notion de "LIBRAIRIE" pouvant contenir des programmes-source ou des textes. L'utilisateur peut alors regrouper en un même fichier indexé les divers éléments d'une application et, ainsi, simplifier la gestion de son espace disque.

EDIT16 permet de traiter les articles d'un fichier indexé dans les mêmes conditions que les fichiers séquentiels. Chaque fois que la forme "nomfic" est possible, la forme "nomartic.nomfic" l'est également, sous réserve que le fichier nommé soit de type indexé. La syntaxe et le fonctionnement de la commande EDIT et des directives G(Get), W(Write) et M(Macro) sont adaptés pour prendre en compte cette nouvelle forme ; le fonctionnement des directives S(Save) et F(File) également.

11.1 COMMANDES ET DIRECTIVES

11.1.1 COMMANDE EDIT (activation d'EDIT16)

Syntaxe :

```
EDIT nomartic.nomfic[-pw] [,Un|,Dn|,En] [,NS] [T]
```

nomartic : nom d'article (1 à 8 caractères alphanumériques ou caractères spéciaux ":", ";" et "<").

nomfic : nom du fichier (1 à 6 caractères idem) qui sera considéré comme un fichier indexé, s'il est précédé d'un nom d'article.

Fonctionnement :

Si un nom d'article est précisé, les fonctions suivantes sont réalisées :

- recherche d'un fichier de nom "nomfic"
- contrôle que ce fichier est de type indexé
- recherche dans ce fichier d'un article de nom "nomartic"

1. Si le fichier existe, et qu'il est de type indexé, l'article nommé est recherché.

- . Si l'article est trouvé, le texte est recopié dans le fichier de travail et l'éditeur se place en environnement EDIT, prêt à accepter des directives.
- . Si l'article n'est pas trouvé, l'éditeur se place

en environnement INPUT, afin que l'utilisateur entre des lignes de texte. Si, ultérieurement, une directive S(Save) ou F(File) est émise l'article sera créé.

2. Si le fichier n'existe pas, il est créé automatiquement et l'éditeur se place en environnement INPUT. Si une directive S ou F est ultérieurement émise, l'article nécessaire à la sauvegarde du texte sera créé. Au cas où il ne sauvegarderait pas son texte, l'utilisateur devrait procéder lui-même à la suppression du fichier indexé qui vient d'être créé.

Messages :

WRONG TYPE	Le fichier n'est pas de type indexé si un nom d'article a été donné (ou n'est pas de type séquentiel dans le cas contraire).
NEW FILE	Le fichier nommé n'existe pas. Un fichier indexé nouveau est créé.
NEW ITEM	L'article nommé n'existe pas. Un nouveau sera créé avec ce nom.

11.1.2 DIRECTIVES S (SAVE) et F(FILE)

Syntaxe identique :

S ou F

Fonctionnement :

Le texte édité est sauvegardé dans l'article de fichier indexé, précisé lors de la commande EDIT.

1. Si l'article existe avant la session, il est renommé, pendant le temps de la création et de l'écriture d'un nouvel article contenant le texte édité. Il est détruit ensuite, si cette opération s'est correctement exécutée. En cas de difficultés en cours d'exécution de la directive, l'article ancien sera retrouvé avec un nom de la forme :

<<JJHHMM

avec JJ représentant le jour du mois
HH " " l'heure
MM " " les minutes.

2. Dans le cas de création d'un article de fichier indexé, l'écriture se réalise nécessairement dans un nouvel article situé en fin de fichier indexé.
3. Dans le cas où le nombre maximum d'article est atteint (fichier saturé), l'utilisateur en est informé. Un fichier séquentiel est créé ; il contient le texte édité. Le nom choisi par l'éditeur pour ce fichier est mentionné. Le contrôle est alors rendu au dialogue du système.

Réponse :

1. nomartic.nomfic-ct
EDIT :

2. OVERFLOW fichier indexé saturé.
<<0123-CT nom du fichier séquentiel créé.
END (voir le paragraphe : saturation de table index)

11.1.1.3 DIRECTIVE W (WRITE)

Syntaxe :

```
W nomartic.nomfic [-ct] [,Un|,Dn|,En] [,NS] n
```

Fonctionnement :

La création d'un article de fichier indexé est demandée.

Si le fichier n'existe pas, il est créé (avec un nombre maximum d'articles de 31) et le texte est recopié après création d'un article.

Si le fichier existe, un article est créé pour recevoir le texte édité. Dans le cas où l'article existe déjà dans ce fichier, la directive est refusée.

Messages :

NEW FILE	un fichier indexé nouveau est créé.
DUPL. NAME	un article de ce nom existe déjà dans le fichier.
WRONG TYPE	Le fichier nommé n'est pas de type indexé.

11.1.1.4 DIRECTIVE G (GET)

Syntaxe :

```
G nomartic.nomfic [-ct] [,Un|,Dn|,En] [,NS] np nd
```

Fonctionnement :

La lecture d'un article est demandée. Celui ci est recherché dans le fichier indexé et les lignes demandées sont insérées derrière la ligne courante.

Messages :

NOT FOUND	le fichier indexé n'existe pas, ou l'article n'existe pas dans le fichier.
WRONG TYPE	le fichier nommé n'est pas de type indexé.

11.1.1.5 DIRECTIVE M (MACRO)

Syntaxe :

```
M nomartic.nomfic [-ct] [,Un|,Dn|,Dn|,En] [,NS]
```

Fonctionnement :

La recherche des définitions de macro est réalisée dans l'article

précisé.

Messages :

NOT FOUND le fichier indexé n'existe pas ou l'article
 n'existe pas dans le fichier.

WRONG TYPE Le fichier nommé n'est pas de type indexé.

11.2 PARTICULARITES D'UTILISATION DES FICHIERS INDEXES

Il appartient à l'utilisateur de veiller à la gestion du fichier indexé ; celle-ci n'est pas prise en compte par EDIT16.

Rappelons que

- Un fichier indexé doit être créé avec mention du nombre maximum d'articles possible. En cas de dépassement de ce nombre, la création d'un article n'est pas possible (saturation de table d'index).
- . La destruction d'un article ne s'accompagne pas de la récupération de l'espace occupé, mais libère une position dans la table d'index.
- . L'écriture d'un nouvel article est toujours réalisée en fin de fichier après le dernier article existant.
- . Un article de fichier indexé ne peut excéder 64K mots, c'est à dire 128K caractères. (N.B. une ligne de n caractères occupe n + 2 caractères).

11.2.1 CREATION DU FICHIER INDEXE

La création du fichier indexé peut être réalisée de trois façons :

1. Par une commande de création émise sous dialogue du système. Elle précisera le nombre maximum (n) d'articles, selon la syntaxe :

```
CREATE nomfic-ct,fu,I(n)
```

2. Par un autre processeur
3. Par EDIT16 lui même, si le fichier indexé nommé dans la commande EDIT ou la directive W n'existe pas. Il est alors créé avec une table suffisante pour 31 articles. L'utilisateur en est informé par le message : NEW FILE. Il appartient à l'utilisateur de le détruire si, par erreur, il a donné un nom de fichier indexé inexistant.

11.2.2 GESTION DU FICHER INDEXE

Cette gestion sera réalisée hors des sessions EDIT16 à l'aide des processeurs utilitaires FUP2 et FUP5 et des commandes suivantes :

Détruire un article :	
Commande	DELETE nomartic.nomfic-ct
ou	FUP5 avec IFSUP
Renommer un article :	FUP5 avec IFRNAM
Augmenter le nombre maximum d'articles :	FUP5 avec IFEXT
Compacter un fichier :	FUP5 avec IFSQUEEZ
Lister les articles	FUP2 avec IFLIST

11.2.3 COMPLEMENTS ET EXEMPLES

Nom de fichier indexé

Si l'édition a été lancée sur un article de fichier indexé, les directives G, W et M pourront, si elles adressent des articles du même fichier, utiliser une forme simplifiée de syntaxe : le nom de fichier indexé est remplacé par le caractère "=". Ce qui signifie : même fichier indexé que lors de la commande EDIT.

Exemple :

```
EDIT ab.biblio
....
M macro.=          équivalent à M macro.biblio
....
W cd.= 52          "      "      W cd.biblio 52
```

Attention : le nom de catalogue n'est pas inclus dans cette forme et devra être fourni, s'il n'a pas été fourni par défaut dans la commande EDIT.

Taille des articles d'un fichier indexé

Un article ne peut excéder la taille de 64K mots soit 128K caractères ainsi :

Pour un texte théorique qui contiendrait des lignes d'égale longueur, on peut calculer le nombre de lignes admissibles avec la formule :

$$NL=131072/(NCPL + 2)$$

NL : nombre maximum de lignes
NCPL : nombre de caractères par ligne

Ainsi pour

NCPL	NL
80	1598
40	3120
30	4096

Si sur une directive S ou F apparaît une erreur du type :

ERE 8 '6017 : article trop long,
l'utilisateur devra détruire, le cas échéant, l'article qui a le nom précisé
à la commande EDIT et renommer l'article qui a un nom de la forme :

```
<<JJHHMM  
(voir directives S et F dans ce chapitre)
```

Exemple :

```
CALL FUP2  
IFLI,LIBR-AA
```

....

```
<<171421 1438 MOTS  
L'article a été renommé le 17ème jour du mois à 14H 21 minutes
```

Action possible :

```
U1 E5  
CALL FUP5  
IFSU,ARTICLE.LIBR-AA,U1  
IFRN,<<171421.LIBR-AA,ARTICLE.LIBR-AA,U1  
IFSQ,LIBR-AA,U1 (éventuellement)  
CALL FUP2  
IFLI,LIBR-AA,U1
```

```
....  
ARTICLE 1438 MOTS
```

Saturation de la table d'index

La taille de la table d'index est déterminée à la création du fichier indexé. Elle détermine le nombre maximum d'articles que peut contenir le fichier indexé.

Rappelons qu'en cas de saturation de cette table (voir directives S et F dans ce chapitre), un fichier séquentiel est créé et un message "OVERFLOW" est émis, suivi du nom choisi pour ce fichier : il a la forme "<<JJHH-CT" Avec la directive W, seul le message OVERFLOW apparaît, suivi d'un ?.

Remarque :

Pour un fichier indexé acceptant un nombre maximum "N" d'articles, il sera prudent de n'utiliser que N-1 articles afin que la technique utilisée par les directives S et F puisse être appliquée.

En cas de saturation on pourra s'inspirer de l'exemple suivant :

Exemple :

```
EDIT 31EME.LIB30A  
.....  
F  
PUBW-7  
31EME.LIB30A-08  
OVERFLOW saturation de table d'index (TIX)  
<<1915-08 un fichier séquentiel de ce nom  
a été créé le 19ème jour à 15 heures  
END
```

Solution :

```
CALL FUP2  
IFLI,LIB30A-08  
  
<<191532 120 MOTS  
l'article édité à été renommé par FILE  
le 19 à 15 heures 32 minutes.  
il contient l'ancien texte avant EDIT  
DELE <<191532.LIB30A-08
```



```
CALL EDIT16  
EDIT 31EME.LIB30A  
NEW ITEM  
INPUT :
```

```
EDIT :
```

```
G <<1915-08 1 10000    GET pour récupérer le texte  
                        sur le fichier séquentiel
```

```
F
```

```
PUBW=7
```

```
31EME.LIB30A-08
```

```
END
```

```
..sans problème car cet article est nouveau  
..Attention : la TIX est pleine.  
                (faire IFEXT avec FUP5)
```

```
.....
```

12 MISE EN OEUVRE DE L'EDITEUR

12.1 CONFIGURATION NECESSAIRE

EDIT16 est un processeur fonctionnant en mode maître ou esclave sous tous les systèmes d'exploitation disque du SOLAR 16.

Il nécessite une partition mémoire minimum de 15 K mots. Une taille mémoire plus importante sera nécessaire en cas d'utilisation de macro-directives.

La taille disque nécessaire est fonction de la taille de l'espace virtuel ; elle est liée au nombre de lignes maximum que peut traiter EDIT16.

En standard, EDIT16 peut gérer un fichier de 5460 lignes, ce qui correspond fichier espace virtuel de l'ordre de 260 K mots, implanté sur l'unité fonctionnelle D2.

La taille du fichier espace virtuel et l'unité fonctionnelle disque qui le supporte, peuvent être modifiées lors de l'intégration de l'éditeur EDIT16.

12.2 INTEGRATION DE L'EDITEUR

EDIT16 est livré sous forme d'une Image Mémoire supportée par le fichier de nom EDIT16-S

Celle-ci est utilisable directement (commande CALL), en mode esclave, avec un espace virtuel standard sur unité fonctionnelle D2, pour éditer des fichiers comprenant en standard jusqu'à 5460 lignes.

12.2.1 MODE MAITRE

Pour une utilisation en mode maître, la création de l'image mémoire appropriée sera réalisée, à partir du fichier EDIT16-BT contenant la forme binaire translatable, au moyen des commandes suivantes sous BOS-G ou BOS-D

```
* DELE EDIT16-S
* CALL BUILD
* BI EDIT16-BT
* MLOD,,16
* CATAL IM,EDIT16-S
* EOJ.
```

12.2.2 ADAPTATION DES PARAMETRES.

Les paramètres "FU support de l'espace virtuel" et "taille de l'espace virtuel" peuvent être modifiés si nécessaire.

Les trois premiers mots de l'éditeur contiennent, en effet, respectivement :

- le numéro d'unité fonctionnelle disque support de l'espace virtuel standard, en hexadécimal

- un indicateur caractérisant la taille de l'espace virtuel standard. Sa

valeur est à

0 pour 1 364 lignes de texte
1 pour 2 729 lignes
2 pour 5 480 lignes
3 pour 10 921 lignes

- un indicateur de valeur 0 à 3 comme le précédent caractérisant la taille maximale des espaces virtuels spécifiques que l'on peut créer par la commande SPACE.

12.2.2.1 Modification de la FU support de l'espace virtuel

Emettre les commandes :

```
CALL FUP7
```

```
PATCH,EDIT16-:S,D2
```

```
ADRE, '0, , ROOT
```

```
VISU
```

```
'000E (en standard D2)
```

```
STOR,'numéro de FU (par exemple STOR, '14 pour D8)
```

La réponse "000E" est émise par le système

Remarque :

Si le numéro de FU est nul (STOR, '0) il n'existera pas d'espace virtuel standard. Chaque activation de l'éditeur (au moyen de la commande CALL) devra être suivie d'une commande SPACE.

12.2.2.2 Modification de la taille de l'espace virtuel standard

Emettre les commandes :

```
CALL FUP7
```

```
PATCH,EDIT16-:S,D2
```

```
ADRE, '1, , ROOT
```

```
VISU
```

```
'0002 (en standard 5460 lignes)
```

```
STOR, '2 (par exemple STOR, '1 pour 2729 lignes).
```

12.2.2.3 Modification de la taille maximum des espaces virtuels spécifiques

Emettre les commandes :

```
CALL FUP7
```

```
PATCH,EDIT16-:S,D2
```

```
ADRE, '2, , ROOT
```

```
VISU
```

```
'0003 (en standard 10921 lignes)
```



STOR, 'i (par exemple STOR,'2 pour 5460 lignes)

12.2.3 INTEGRATION SOUS LES SYSTEMES TSM et TSF/MPES

EDIT16 est intégré sous les systèmes TSM et TSF en tant qu'utilitaire. Les fichiers espaces virtuels devront obligatoirement se trouver dans des unités fonctionnelles disques banales.

Il est conseillé par ailleurs de ne pas utiliser l'unité fonctionnelle disque D2 comme support des espaces virtuels (encombrement ; D2 réservé aux systèmes).

Il convient donc de modifier le numéro de la FU support des espaces virtuels.

D'autre part, pour minimiser l'encombrement des espaces virtuels, il suffit d'initialiser sur le disque, la FU support des espaces virtuels à 0. Chaque utilisateur devra alors avant d'éditer un fichier, émettre une commande SPACE dans laquelle il précisera la taille maximum du fichier à éditer et le support de l'espace virtuel (le support pouvant être dès lors soit une FU banale, soit une FU volume, soit une FU privée). La durée de vie d'un tel espace virtuel est limitée à la durée de l'édition d'un fichier.

12.2.4 CONTRAINTES FMS

La taille des granules de l'unité fonctionnelle disque support d'un espace virtuel doit respecter le tableau suivant :

Valeur de i	Nombre de lignes	Nombre d'articles de 1 Kmots dans l'espace virtuel	Taille minimum des granules (en secteurs)
0	1364	65	8
1	2729	129	16
2	5460	257	32
3	10921	513	48

13 ANNEXE

13.1 SOMMAIRE COMMANDES ET DIRECTIVES

13.1.1 COMMANDES D'ACTIVATION

COMMANDES D'ACTIVATION	SYNTAXE
Lancement d'édition	EDIT nomfic[-ct] [,Ui ,Di ,Ei] [,NS] [T]
" sur article	EDIT nomartic.nomfic.
Retour avec reprise	GO
Retour avec abandon	STOP
Définition espace spécifiq	SPACE i,Un

13.1.2 DIRECTIVES DE SAUVEGARDE ET FIN DE SESSION

SIGNIFICATION	SYNTAXE	ZONE
Sauvegarde et reprise édition (SAVE)	S	
Sauvegarde et fin d'édition (FILE)	F	
(" " enregistrements 80 caractères)	F F80	
Fin d'édition sans sauvegarde (QUIT)	Q	

SIGNIFICATION	SYNTAXE	ZONE
Positionnement en avant (NEXT) <ul style="list-style-type: none"> - relatif ligne courante - contexte simple - contextes multiples 	N [n] N /chaîne/ N (* +)/chl/ . . . /Chi/	 Z Z
Positionnement en arrière (UP) <ul style="list-style-type: none"> - relatif - contexte simple - contextes multiples 	U [N] U /chaîne/ U (* +)/chl.../chi/	 Z Z
Positionnement dernière Ligne (BOTTOM)	B	
Positionnement ligne zéro (TOP)	T	
Positionnement absolu (JUMP)	J [n]	
Présentation numéro logique (LINE)	L	
Présentation numéro absolu (ORDER)	Q	
Répétition de direct. précédente (AGAIN)	A [n]	

13.1.4 DIRECTIVES SUR CONTENU DE LIGNE

SIGNIFICATION	SYNTAXE	ZONE
Remplacement de chaîne (CHANGE)	C /ch1/ch2/ [n *] [n *]	Z
Recherche colonnes de-(KEY) - de début de chaîne	K /ch/	Z
- des limites du texte	K	
- de fin de ligne	K /	
Présentation de chaîne	X nc [l] [nl *]	Z
Insertion de chaîne	Y [nc] /ch/ [nl *]	Z
Impression de lignes (PRINT)		
- sur console de dialogue	P [nl]	Z
- sur unité d'impression LO	P:[nl] [P] [T] [N]	Z
Définition de zone	Z np nd [I]	
Annulation de zone	Z	
Présentation de zone	Z?	
Equivalence zone - champ	Z chp [I]	
Modification séparateur de chaîne	/car	
Caractère non imprimable	H car ['XX]	

13.1.5 DIRECTIVES SUR LIGNES

SIGNIFICATION		CHAMP	ZONE
Suppression de lignes (DELETE)	D [n]		
Insertion de lignes (INSERT)			
- immédiate	I ligne	A-Y	Z
- en mode INPUT	I	A-Y	Z
- depuis unité lecture SI	I: [n]	A-Y	Z
Remplacement de ligne (REPLACE)			
- immédiate	R ligne	A-Y	Z
- en mode INPUT	R	A-Y	Z
- depuis unité lecture SI	R: [n]	A-Y	Z
Definition tabulation et champs	E n1 n2 ..ni [NT c]		
Présentation definition de champs	E?		
Annulation de definition	E		
Modification caractère tabulation	>car		

13.1.6 DIRECTIVES D'ACCES FICHIERS et MACRO DIRECTIVES

SIGNIFICATION	SYNTAXE	Z O N E
Transfert vers autre fichier (WRITE)	W ndf.. (n *) [R E] [F80 Fnf]	Z
Insertion depuis autre fichier (GET)	G ndf.. np nd	Z
Création macro-definitions (MACRO) <ul style="list-style-type: none"> - depuis fichier externe - sur console de dialogue Appel Macro-directive	M ndf.. M %macro	
	avec ndf ..=[nomertic.] nomfic [-ct] [,Ui ,Di ,Ei] [,NS]	

13.2 SOMMAIRE ECRITURE DE MACRO-DIRECTIVES

INSTRUCTION DE MACRO-DIRECTIVE	SIGNIFICATION
<p>*DEFentete (avec [?] ,max 9) *ENDEF *Vnn = expr. arithm. (+ - * /) *Vnn EQ expr. chaîne *SKIP expr. arithm. *IF expr. (< = >) expr. SKIP expr. *IF expr. (EQ NE) expr. SKIP expr. *KILL texte *PRINT texte *LPRINT texte *REPLY [n] *T Liste ligne</p>	<p>Définition de macrodirective Fin de définition Affectation variable arithm. Affectation variable chaîne Saut inconditionnel Saut conditionnel (arithm.) Saut conditionnel (chaîne) Abandon macro-génération Impression ligne de dialogue Impression TEXTE sans fin ligne Lecture réponse dialogue Tabulation Génération ligne</p>
<p>?P1 à ?P9 : valeur paramètre . ?Ln : longueur . ?Tn : type 1 à 5 @V01 à @V99 : valeur de variable définie par *Vnn ?P0 : valeur rendue par directive d'édition ou *REPLY @V00 : 1 si directive bien exécutée, 0 si anomalie ?Hxx : caractère non imprimable en hexadécimal</p>	

13.3 MESSAGES

13.3.1 MESSAGES D'INFORMATION

	SIGNIFICATION
NEW FILE	Le fichier à éditer n'existe pas. Il sera créé.
INPUT :	Environnement INPUT. Entrée de lignes.
EDIT :	Environnement EDIT. Entrée de directives
TOP	Début de Texte : ligne zéro
EOF	Fin de Texte : dernière ligne
DEFMAC :	Environnement MACRO-DEFINITION. Entrée de définitions
PUBW = n	Numéro de catalogue des fichiers de travail d'EDIT 16 si utilisation multiconsole.
nomfic-ct	Nom et catalogue du fichier en cours de sauvegarde.
END	Fin normale de la session EDIT16
IMPLICIT Z	Définition de zone implicitement utilisée
NEW FILE	Le fichier indexé n'existait pas. Il est créé.
NEW ITEM	L'article à éditer n'existe pas. Il sera créé.

13.3.2 MESSAGES D'ANOMALIES

MESSAGE	SIGNIFICATION	ACTION EVENTUELLE
?	Syntaxe incorrecte, la directive n'est pas exécutée	Réémettre la directive
TRUNCATED	Perte de caractères en fin de ligne ou de zone	
NO CHANGE	Directive C : chaîne non trouvée	
NO LINE	Directive J : ligne inexistante	
END LINE	Directive K : chaîne non trouvée	
NO ZONE	Zone non définie, ou annulée	Emettre une directive Z
NO FIELD	Aucun champ n'est défini	Emettre une directive E
UNDEFINED	Ce champ n'existe pas	Réémettre directive E
PARITY ERR	Erreur d'entrée clavier. La ligne est ignorée	Refrapper la ligne
TIME OUT	" " " " " "	" " "
NO SPACE	Il n'y a pas d'espace standard	Emettre une commande SPACE
TOO LINES	texte trop grand pour l'espace défini	" " "
WRONG TYPE	Le fichier nommé n'est pas de type approprié : séquentiel ou indexé	Réémettre
IN USE	Le fichier nommé est ouvert	
OVERFLOW	Directive S F ou W : le fichier indexé est saturé. Un fichier séquentiel a été créé, si S ou F	Utiliser FUP5
NOT FOUND	Directive G ou M : l'article n'a pas été trouvé dans le fichier indexé.	Réémettre
DUPL.NAME	Directive W. Un article de ce nom existe déjà	Réémettre

13.3.3 MESSAGES D'ERREURS FATALES : ERE n

MESSAGE	SIGNIFICATION	CORRECTION
ERE 0	Commande EDIT erronée	Emettre une nouvelle commande EDIT
ERE 1	Chaînage des lignes incorrect dans l'espace virtuel à la suite d'une erreur de matériel.	Reprendre l'édition à partir du fichier <<TE<< après l'avoir renommé.
ERE 2	Fichier à éditer trop long	Emettre une commande SPACE appropriée au fichier à éditer
ERE 3	Erreur lors de l'ouverture du fichier à éditer	Correction correspondant au défaut détecté par FMS
ERE 4	Erreur lors de la création du fichier <<TE<<. Il existe déjà.	Détruire ou renommer le fichier <<TE<<
ERE 5	Erreur lors de l'ouverture du fichier de travail <<VI<<	Correction correspondant au défaut détecté par FMS
ERE 6	Enchaînement de commandes incorrect (GO ou STOP avant EDIT)	
ERE 7	Erreur lors d'un accès au fichier de travail : espace virtuel : <<VI<<	Correction selon défaut détecté par FMS. Refaire l'édition à partir du fichier <<TE<< s'il existe, ou du fichier initial.
ERE 8	Erreur détectée par FMS . commande EDIT : lors de la lecture du fichier à éditer . directive S ou F : lors de la recopie du fichier édité . directive G W ou M : lors de l'accès au fichier désigné	Correction selon défaut détecté par FMS.
ERE 9	Défaut apparu ou BREAK émis sur le périphérique de dialogue Défaut sur périphérique associé aux unités symboliques SI ou L0	Correction ou action, selon le cas, puis reprise de l'édition par commande GO ou STOP.
ERE 10	Erreur détectée par FMS lors d'un accès au fichier indexé	Correction selon défaut constaté
<p>*NOTE : Pour les ERE 3, 4, 5, 7, 8 et 10, le numéro d'erreur FMS est fourni en information complémentaire.</p> <p>Pour l'ERE 9, le compte-rendu IOCS est fourni.</p>		

13.3.4 MESSAGES D'ERREURS MACRO-DIRECTIVES : ERMnn

MESSAGE	SIGNIFICATION
ERM 00	Erreur de parité
ERM 01	Erreur d'écriture - Erreur dans l'écriture d'une instruction de macro - Numéro de variable supérieur à 99
ERM 02	Expression arithmétique incorrecte - Opérande non décimal - Enchaînement d'opérateurs interdit - Division par zéro - Valeur d'un opérande non représentable sur 16 bits - Valeur d'une expression non représentable sur 16 bits - etc.....
ERM 03	Contexte incorrect - ENDEF, KILL, SKIP, IF... hors d'une macro-définition
ERM 04	Élément non défini : variable ou paramètre
ERM 05	Appel de macro incorrect ou définition non intégrée
ERM 06	Definitions imbriquées
ERM 09	Saut à l'extérieur d'une macro
ERM 11	Niveau d'imbrication d'une macro supérieur à 10
ERM 12	Saturation des tables. Espace libre dans la partition insuffisant
ERM 14	Ligne générée trop longue
ERM 15	Saturation de l'espace alloué aux variables



Distribution codes/Codes de diffusion			
Customers : Clients :			
Internal : Interne :			

DELIVERY ADDRESS
ÉTIQUETTE ADRESSE

Bull MTS

1, Rue de Provence
B. P. 208
38432 ÉCHIROLLES CEDEX / FRANCE



Sems