

# **Gestion des travaux et IOF**

## **Bull DPS 7000**

### **JCL - Manuel de référence**

#### **GCOS 7**

---

##### **Logiciel**

**Sujet :** Description du JCL utilisé sous GCOS 7.

**Observations :** La révision 4 annule et remplace la révision précédente pour tous les utilisateurs de GCOS 7-V8.

**Version du logiciel :** GCOS 7-V8 TS 8560

**Date :** Mars 1998

**BULL ELECTRONICS ANGERS S.A.**  
**CEDOC**  
Atelier de reprographie  
331, Avenue Patton  
49004 ANGERS Cedex 01  
FRANCE

**BULL HN INFORMATION SYSTEMS INC.**  
Publication Order Entry  
Fax: (508) 294-7411  
MA02/423S  
Technology Park  
Billerica, MA 01821  
U.S.A.

Copyright © Bull S.A., 1991, 1996, 1997, 1998

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

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

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

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

# Préface

## OBJET DU MANUEL

Ce manuel décrit les ordres JCL utilisés sous le système d'exploitation GCOS 7.

## UTILISATEURS CONCERNES

Ce manuel est destiné à tous les utilisateurs des systèmes DPS 7000 fonctionnant sous GCOS 7 V8.

Il ne demande qu'une connaissance générale de GCOS et la connaissance des principes de la gestion des travaux.

## STRUCTURE DU MANUEL

Chapitre 1	Contient une présentation générale de la gestion des travaux et du rôle du langage JCL sous GCOS, ainsi qu'une explication de la différence entre ordres JCL de base (fonctions de base du système) et ordres JCL étendus (appels d'utilitaires).
Chapitre 2	Expose les règles d'écriture des ordres JCL.
Chapitre 3	Explique comment désigner les fichiers, les volumes et les appareils dans les ordres JCL.
Chapitre 4	Décrit de façon détaillée tous les ordres JCL de base, qui sont classés par ordre alphabétique.
Chapitre 5	Donne la liste des ordres JCL étendus avec une brève explication et une référence bibliographique.

Annexe A	Donne la liste de tous les ordres JCL dotés d'une abréviation.
Annexe B	Contient des informations sur la conversion ASCII-EBCDIC des caractères.
Glossaire	Définit certains termes rencontrés dans ce manuel.
Index	Facilite la recherche.

## CONVENTIONS D'ECRITURE

Les conventions d'écriture utilisées dans ce manuel pour la syntaxe des ordres JCL sont les suivantes :

ELEMENT	Les majuscules indiquent un nom ou un mot-clé, à introduire tel quel.
élément	Les minuscules indiquent une valeur à fournir par l'utilisateur.
[élément]	Les crochets indiquent un élément facultatif.
{élément1} {élément2} {élément3}	Une colonne d'éléments entre accolades indique que l'un d'eux doit être sélectionné lorsque le paramètre correspondant est retenu.
élément	Un élément souligné représente la valeur implicite. C'est la valeur que le système retient lorsqu'aucune autre n'est spécifiée.
()	Les parenthèses sont obligatoires lorsqu'elles renferment plus d'une valeur positionnelle.
élément...	Les points de suspension indiquent que l'élément peut être répété une ou plusieurs fois.

Les formats d'ordres apparaissent dans une police différente de celle du texte.

## BIBLIOGRAPHIE

La lecture de ce manuel sera utilement complétée par celle du *Manuel utilisateur JCL*. Le lecteur pourra également se reporter au *Répertoire des messages d'erreur et codes retour*.

Les manuels dont la référence est donnée ci-après (et également aux endroits appropriés du texte) permettent d'obtenir davantage d'informations sur un sujet précis.

### Communications

<i>GCOS 7 MCS - Guide de l'utilisateur</i> .....	47 F2 32UC
<i>GCOS 7-V3A7/V3B7/V5 MCS User's Guide</i> .....	47 A2 32UC
<i>Fonctions à distance DPS 6/DPS 7</i> .....	47 F2 11UC
<i>Remote Facilities DPS 7 to DPS 6</i> .....	47 A2 11UC
<i>UFT - Guide de l'utilisateur</i> .....	47 F2 13UC
<i>Unified File Transfer (UFT) User's Guide</i> .....	47 A2 13UC
<i>DJP - Guide de l'utilisateur</i> .....	47 F2 14UC
<i>Distributed Job Processing (DJP) User's Guide</i> .....	47 A2 14UC

### Gestion des fichiers et utilitaires

<i>UFAS-EXTENDED - Guide de l'utilisateur</i> .....	47 F2 04UF
<i>UFAS-EXTENDED User's Guide</i> .....	47 A2 04UF
<i>GAC-EXTENDED - Guide de l'utilisateur</i> .....	47 F2 11UF
<i>GAC-EXTENDED User's Guide</i> .....	47 A2 11UF
<i>Utilitaires de gestion de données</i> .....	47 F2 26UF
<i>Data Management Utilities</i> .....	47 A2 26UF
<i>Gestion des catalogues - Guide de l'utilisateur</i> .....	47 F2 35UF
<i>Catalog Management User's Guide</i> .....	47 A2 35UF
<i>Cartridge Tape Library Unix Server User's Guide</i> .....	47 A2 63UU

### Fonctions interactives

<i>FORMS - Guide de l'utilisateur</i> .....	47 F2 15US
<i>FORMS User Guide</i> .....	47 A2 15US
<i>GCL - Guide du programmeur</i> .....	47 F2 36UJ
<i>GCL Programmer's Manual</i> .....	47 A2 36UJ
<i>IOF - Manuel de référence de l'utilisateur</i>	
<i>Vol. 1 : Introduction à IOF</i> .....	47 F2 38UJ
<i>Vol. 2 : Commandes GCL</i> .....	47 F2 39UJ
<i>Vol. 3 : Commandes processeurs</i> .....	47 F2 40UJ

IOF Terminal User's Reference Manual	
Part 1 : Introduction to IOF.....	47 A2 38UJ
Part 2 : GCL Commands.....	47 A2 39UJ
Part 3 : Processors Commands .....	47 A2 40UJ

### Langages

COBOL 85 - Guide de l'utilisateur .....	47 F2 06UL
COBOL 85 User's Guide.....	47 A2 06UL
FORTRAN 77 - Guide de l'utilisateur .....	47 F2 16UL
FORTRAN 77 User's Guide .....	47 A2 16UL
GPL - Guide de l'utilisateur .....	47 F2 36UL
GPL User's Guide .....	47 A2 36UL
PASCAL - Guide de l'utilisateur .....	47 F2 52UL
PASCAL User's Guide .....	47 A2 52UL

### Préparation des programmes

Maintenance des bibliothèques - Manuel de référence.....	47 F2 01UP
Library Maintenance Reference Manual .....	47 A2 01UP
Maintenance des bibliothèques - Guide de l'utilisateur .....	47 F2 02UP
Library Maintenance User's Guide .....	47 A2 02UP
FSE (Editeur plein écran) - Guide de l'utilisateur .....	47 F2 06UP
Full Screen Editor User's Guide .....	47 A2 06UP
PCF (Aide à la mise au point) - Guide de l'utilisateur.....	47 F2 15UP
Program Checkout Facility User's Guide .....	47 A2 15UP

### Administration du système

GCOS 7-V8 - Manuel de l'administrateur du système .....	47 F2 54US
GCOS 7-V8 System Administrator's Manual.....	47 A2 54US
GCOS 7-V8 - Manuel d'exploitation .....	47 F2 53US
GCOS 7-V8 System Operator's Guide.....	47 A2 53US

### Sous-système transactionnel TDS

TDS - Guide de l'administrateur.....	47 F2 22UT
TDS Administrator's Guide.....	47 A2 22UT
TDS/COBOL - Manuel du programmeur.....	47 F2 23UT
TDS/COBOL Programmer's Manual .....	47 A2 23UT

# Table des matières

<b>1.</b>	<b>Introduction .....</b>	1-1
<b>1.1</b>	<b>DESCRIPTION D'UN TRAVAIL.....</b>	1-1
1.1.1	<b>Terminologie.....</b>	1-1
1.1.2	<b>Rubrique de description de travail.....</b>	1-1
1.1.3	<b>Rubrique de description d'activité .....</b>	1-3
1.1.4	<b>Rubrique de description d'entrées.....</b>	1-3
1.1.5	<b>Rubrique de description de données.....</b>	1-3
1.1.6	<b>Exemple de train d'entrées.....</b>	1-3
<b>1.2</b>	<b>TYPES D'ORDRES JCL.....</b>	1-5
<b>1.3</b>	<b>SOUMISSION D'UN TRAVAIL .....</b>	1-6
<b>1.4</b>	<b>PASSAGE D'UN TRAVAIL ET ETATS D'UN TRAVAIL .....</b>	1-6
<b>1.5</b>	<b>LECTEUR DE TRAVAUX .....</b>	1-9
1.5.1	<b>Lecteur des entrées .....</b>	1-9
1.5.2	<b>Train d'entrées.....</b>	1-10
1.5.3	<b>Traducteur JCL.....</b>	1-11
<b>1.6</b>	<b>PLANIFICATION DES TRAVAUX.....</b>	1-12
<b>1.6.1</b>	<b>Planificateur de travaux.....</b>	1-12
1.6.1.1	<b>File d'attente du planificateur de travaux.....</b>	1-12
<b>1.6.2</b>	<b>Gestion des travaux .....</b>	1-12
<b>1.6.3</b>	<b>Classes de travaux.....</b>	1-13
1.6.3.1	<b>Classes de travaux.....</b>	1-13
1.6.3.2	<b>Attributs d'une classe de travaux .....</b>	1-14

<b>1.7</b>	<b>EXECUTION DES TRAVAUX.....</b>	1-17
1.7.1	<b>Sélection des travaux .....</b>	1-17
1.7.2	<b>Interpréteur d'ordres .....</b>	1-17
1.7.3	<b>Exécution d'une activité .....</b>	1-17
1.7.4	<b>Fin d'activité .....</b>	1-17
1.7.5	<b>Fin de travail .....</b>	1-18
<b>1.8</b>	<b>TRAVAUX DE SERVICE .....</b>	1-20
<b>1.9</b>	<b>FONCTIONS ACCESSIBLES AU MOYEN D'ORDRES JCL .....</b>	1-22
1.9.1	<b>Désynchronisation des E/S sur périphériques autres que disques et bandes...</b>	1-22
<b>1.10</b>	<b>GESTION DES ENTREES.....</b>	1-23
1.10.1	<b>Mécanisme SYSIN .....</b>	1-23
1.10.2	<b>Utilisation du mécanisme SYSIN .....</b>	1-23
<b>1.11</b>	<b>GESTION DES SORTIES .....</b>	1-24
1.11.1	<b>Editeur des sorties .....</b>	1-24
1.11.2	<b>Fichier SYSOUT standard (SYS.OUT) .....</b>	1-24
1.11.3	<b>Fichiers SYSOUT permanents .....</b>	1-24
1.11.4	<b>Mécanisme SYSOUT et format du fichier SYSOUT .....</b>	1-25
<b>1.12</b>	<b>JCL ENREGISTRÉ .....</b>	1-26
1.12.1	<b>Mini-éditeur .....</b>	1-26
1.12.2	<b>Références aux paramètres dans les ordres JCL.....</b>	1-26
<b>1.13</b>	<b>PARTAGE ET CONSERVATION DE FICHIERS .....</b>	1-27
<b>1.14</b>	<b>UTILISATION DES COMMANDES OPÉRATEUR.....</b>	1-28
<b>1.15</b>	<b>RELANCE SUR POINT DE REPRISE.....</b>	1-28
<b>1.16</b>	<b>JOURNALISATION .....</b>	1-28
<b>2.</b>	<b>Règles d'écriture des ordres JCL .....</b>	2-1
<b>2.1</b>	<b>GENERALITES.....</b>	2-1

## Table des matières

<b>2.2</b>	<b>ELEMENTS D'UN ORDRE JCL .....</b>	2-3
2.2.1	<b>Etiquette .....</b>	2-3
2.2.2	<b>Nom d'ordre .....</b>	2-3
2.2.3	<b>Paramètres .....</b>	2-4
2.2.3.1	Types de paramètres .....	2-4
2.2.3.2	Valeurs implicites des paramètres .....	2-7
2.2.3.3	Chaînes protégées .....	2-7
<b>3.</b>	<b>Identification des fichiers et groupes de paramètres standard .....</b>	3-1
<b>3.1</b>	<b>IDENTIFICATION DES FICHIERS .....</b>	3-1
3.1.1	<b>Types de fichiers .....</b>	3-1
3.1.1.1	Fichiers catalogués, non catalogués et temporaires .....	3-2
3.1.1.2	Fichiers bibliothèque .....	3-2
3.1.1.3	Fichier SYS.IN .....	3-2
3.1.2	<b>Noms de fichiers .....</b>	3-3
3.1.2.1	Noms de fichiers permanents .....	3-3
3.1.2.2	Noms de fichiers temporaires .....	3-7
3.1.2.3	Noms de fichiers bibliothèques et d'unités de bibliothèque .....	3-7
3.1.2.4	Noms de rubriques de description d'entrées .....	3-7
3.1.2.5	Noms de fichiers non standard .....	3-8
3.1.2.6	Rappel .....	3-8
3.1.3	<b>Statut des fichiers .....</b>	3-9
3.1.4	<b>Identification des volumes .....</b>	3-9
3.1.4.1	Volume résident .....	3-9
3.1.4.2	Volume non résident .....	3-10
3.1.4.3	Noms de volumes non standard .....	3-10
3.1.5	<b>Identification des appareils .....</b>	3-11
3.1.5.1	Classe et attributs des appareils .....	3-11
3.1.5.2	Liste des classes et attributs d'appareils existants .....	3-13
3.1.5.3	Nom d'appareil .....	3-16
<b>3.2</b>	<b>GROUPES DE PARAMETRES STANDARD .....</b>	3-17
3.2.1	<b>Description-fichier-d'entrée-séquentiel .....</b>	3-19
3.2.2	<b>Description-fichier-d'entrée .....</b>	3-20
3.2.3	<b>Description-fichier-de-sortie-séquentiel .....</b>	3-20
3.2.4	<b>Description-fichier-de-sortie .....</b>	3-21
3.2.5	<b>Description-fichier-d'impression .....</b>	3-21
3.2.6	<b>Description-fichier-de-travail .....</b>	3-21
3.2.7	<b>Description-bibliothèque-d'entrée .....</b>	3-22
3.2.8	<b>Description-bibliothèque-de-sortie .....</b>	3-23
3.2.9	<b>Description-bibliothèque-d'impression .....</b>	3-23
3.2.10	<b>Paramètres-define .....</b>	3-24
3.2.11	<b>Paramètres-sysout .....</b>	3-24
3.2.12	<b>Paramètres-size .....</b>	3-24

<b>3.2.13</b>	<b>Paramètres-step .....</b>	<b>3-24</b>
<b>4.</b>	<b>Ordres JCL de base .....</b>	<b>4-1</b>
<b>4.1</b>	<b>PRESENTATION DES ORDRES .....</b>	<b>4-1</b>
4.1.1	<b>Fonction .....</b>	4-1
4.1.2	<b>Rubrique .....</b>	4-1
4.1.3	<b>Format .....</b>	4-2
4.1.4	<b>Description .....</b>	4-2
4.1.5	<b>Paramètres .....</b>	4-2
4.1.6	<b>Exemples .....</b>	4-2
<b>4.2</b>	<b>ALLOCATE .....</b>	<b>4-3</b>
4.2.1	<b>Fonction .....</b>	4-3
4.2.2	<b>Rubrique .....</b>	4-3
4.2.3	<b>Format .....</b>	4-3
4.2.4	<b>Description de l'ordre .....</b>	4-4
4.2.5	<b>Paramètres .....</b>	4-5
4.2.5.1	Paramètre obligatoire .....	4-5
4.2.5.2	Paramètres facultatifs .....	4-5
<b>4.3</b>	<b>ASSIGN .....</b>	<b>4-6</b>
4.3.1	<b>Fonction .....</b>	4-6
4.3.2	<b>Rubrique .....</b>	4-6
4.3.3	<b>Format .....</b>	4-6
4.3.4	<b>Description .....</b>	4-8
4.3.5	<b>Paramètres .....</b>	4-13
4.3.5.1	Paramètres obligatoires .....	4-13
4.3.5.2	Paramètres facultatifs .....	4-14
<b>4.4</b>	<b>COMMENT .....</b>	<b>4-30</b>
4.4.1	<b>Fonction .....</b>	4-30
4.4.2	<b>Rubrique .....</b>	4-30
4.4.3	<b>Format .....</b>	4-30
4.4.4	<b>Description .....</b>	4-30
4.4.5	<b>Paramètres .....</b>	4-30
<b>4.5</b>	<b>CONSOLE .....</b>	<b>4-31</b>
4.5.1	<b>Fonction .....</b>	4-31
4.5.2	<b>Rubrique .....</b>	4-31
4.5.3	<b>Format .....</b>	4-31
4.5.4	<b>Description .....</b>	4-31

## Table des matières

<b>4.5.5</b>	<b>Paramètre facultatif</b> .....	4-31
<b>4.6</b>	<b>\$DATA</b> .....	4-32
<b>4.6.1</b>	<b>Fonction</b> .....	4-32
<b>4.6.2</b>	<b>Rubrique</b> .....	4-32
<b>4.6.3</b>	<b>Format</b> .....	4-32
<b>4.6.4</b>	<b>Description</b> .....	4-33
<b>4.6.5</b>	<b>Paramètres</b> .....	4-34
<b>4.6.5.1</b>	Paramètres obligatoires .....	4-34
<b>4.6.5.2</b>	Paramètres facultatifs .....	4-34
<b>4.7</b>	<b>DEFINE</b> .....	4-38
<b>4.7.1</b>	<b>Fonction</b> .....	4-38
<b>4.7.2</b>	<b>Rubrique</b> .....	4-38
<b>4.7.3</b>	<b>Format</b> .....	4-38
<b>4.7.4</b>	<b>Description</b> .....	4-41
<b>4.7.5</b>	<b>Paramètres</b> .....	4-42
<b>4.7.5.1</b>	Paramètre obligatoire .....	4-42
<b>4.7.5.2</b>	Paramètres facultatifs .....	4-42
<b>4.8</b>	<b>\$ENDDATA</b> .....	4-53
<b>4.8.1</b>	<b>Fonction</b> .....	4-53
<b>4.8.2</b>	<b>Rubrique</b> .....	4-53
<b>4.8.3</b>	<b>Format</b> .....	4-53
<b>4.8.4</b>	<b>Description</b> .....	4-53
<b>4.9</b>	<b>\$SENDINPUT</b> .....	4-54
<b>4.9.1</b>	<b>Fonction</b> .....	4-54
<b>4.9.2</b>	<b>Rubrique</b> .....	4-54
<b>4.9.3</b>	<b>Format</b> .....	4-54
<b>4.9.4</b>	<b>Description</b> .....	4-54
<b>4.9.5</b>	<b>Paramètre facultatif</b> .....	4-54
<b>4.10</b>	<b>\$ENDJOB</b> .....	4-55
<b>4.10.1</b>	<b>Fonction</b> .....	4-55
<b>4.10.2</b>	<b>Rubrique</b> .....	4-55
<b>4.10.3</b>	<b>Format</b> .....	4-55
<b>4.10.4</b>	<b>Description</b> .....	4-55
<b>4.11</b>	<b>ENDSTEP</b> .....	4-56
<b>4.11.1</b>	<b>Fonction</b> .....	4-56
<b>4.11.2</b>	<b>Rubrique</b> .....	4-56
<b>4.11.3</b>	<b>Format</b> .....	4-56

<b>4.11.4</b>	<b>Description</b> .....	4-56
<b>4.12</b>	<b>EXDIR</b> .....	4-57
<b>4.12.1</b>	<b>Fonction</b> .....	4-57
<b>4.12.2</b>	<b>Rubrique</b> .....	4-57
<b>4.12.3</b>	<b>Format</b> .....	4-57
<b>4.12.4</b>	<b>Description</b> .....	4-57
<b>4.13</b>	<b>EXECUTE</b> .....	4-58
<b>4.13.1</b>	<b>Fonction</b> .....	4-58
<b>4.13.2</b>	<b>Rubrique</b> .....	4-58
<b>4.13.3</b>	<b>Format</b> .....	4-58
<b>4.13.4</b>	<b>Description</b> .....	4-58
<b>4.13.5</b>	<b>Paramètres</b> .....	4-60
4.13.5.1	Paramètre obligatoire .....	4-60
4.13.5.2	Paramètres facultatifs .....	4-60
<b>4.14</b>	<b>\$INPUT</b> .....	4-62
<b>4.14.1</b>	<b>Fonction</b> .....	4-62
<b>4.14.2</b>	<b>Rubrique</b> .....	4-62
<b>4.14.3</b>	<b>Format</b> .....	4-62
<b>4.14.4</b>	<b>Description</b> .....	4-63
<b>4.14.5</b>	<b>Paramètres</b> .....	4-66
4.14.5.1	Paramètre obligatoire .....	4-66
4.14.5.2	Paramètres facultatifs .....	4-66
<b>4.15</b>	<b>INVOKE</b> .....	4-73
<b>4.15.1</b>	<b>Fonction</b> .....	4-73
<b>4.15.2</b>	<b>Rubrique</b> .....	4-73
<b>4.15.3</b>	<b>Format</b> .....	4-73
<b>4.15.4</b>	<b>Description</b> .....	4-73
<b>4.15.5</b>	<b>Paramètres</b> .....	4-74
4.15.5.1	Paramètres obligatoires .....	4-74
4.15.5.2	Paramètres facultatifs .....	4-74
<b>4.16</b>	<b>\$JOB</b> .....	4-77
<b>4.16.1</b>	<b>Fonction</b> .....	4-77
<b>4.16.2</b>	<b>Rubrique</b> .....	4-77
<b>4.16.3</b>	<b>Format</b> .....	4-77
<b>4.16.4</b>	<b>Description</b> .....	4-78
<b>4.16.5</b>	<b>Paramètres</b> .....	4-78
4.16.5.1	Paramètres obligatoires .....	4-78
4.16.5.2	Paramètres facultatifs .....	4-79

## Table des matières

<b>4.17</b>	<b>JUMP</b> .....	4-84
4.17.1	<b>Fonction</b> .....	4-84
4.17.2	<b>Rubrique</b> .....	4-84
4.17.3	<b>Format</b> .....	4-84
4.17.4	<b>Description</b> .....	4-84
4.17.5	<b>Paramètres</b> .....	4-85
4.17.5.1	Paramètres obligatoires .....	4-85
4.17.5.2	Paramètres facultatifs .....	4-86
<b>4.18</b>	<b>LET</b> .....	4-91
4.18.1	<b>Fonction</b> .....	4-91
4.18.2	<b>Rubrique</b> .....	4-91
4.18.3	<b>Format</b> .....	4-91
4.18.4	<b>Description</b> .....	4-91
4.18.5	<b>Paramètres obligatoires</b> .....	4-92
<b>4.19</b>	<b>LIB</b> .....	4-93
4.19.1	<b>Fonction</b> .....	4-93
4.19.2	<b>Rubrique</b> .....	4-93
4.19.3	<b>Format</b> .....	4-93
4.19.4	<b>Description</b> .....	4-94
4.19.5	<b>Paramètres</b> .....	4-95
4.19.5.1	Paramètre obligatoire .....	4-95
4.19.5.2	Paramètres facultatifs .....	4-95
<b>4.20</b>	<b>MESSAGE</b> .....	4-97
4.20.1	<b>Fonction</b> .....	4-97
4.20.2	<b>Rubrique</b> .....	4-97
4.20.3	<b>Format</b> .....	4-97
4.20.4	<b>Description</b> .....	4-97
<b>4.21</b>	<b>MODVL</b> .....	4-98
4.21.1	<b>Fonction</b> .....	4-98
4.21.2	<b>Rubrique</b> .....	4-98
4.21.3	<b>Format</b> .....	4-98
4.21.4	<b>Description</b> .....	4-98
4.21.5	<b>Paramètres facultatifs</b> .....	4-99
<b>4.22</b>	<b>OUTVAL</b> .....	4-101
4.22.1	<b>Fonction</b> .....	4-101
4.22.2	<b>Rubrique</b> .....	4-101
4.22.3	<b>Format</b> .....	4-101
4.22.4	<b>Description</b> .....	4-102
4.22.5	<b>Paramètres</b> .....	4-102

<b>4.23</b>	<b>POOL</b> .....	4-103
4.23.1	<b>Fonction</b> .....	4-103
4.23.2	<b>Rubrique</b> .....	4-103
4.23.3	<b>Format</b> .....	4-103
4.23.4	<b>Description</b> .....	4-103
4.23.5	<b>Paramètres</b> .....	4-104
4.23.5.1	Paramètres obligatoires .....	4-104
4.23.5.2	Paramètres facultatifs .....	4-105
<b>4.24</b>	<b>PREFIX</b> .....	4-106
4.24.1	<b>Fonction</b> .....	4-106
4.24.2	<b>Rubrique</b> .....	4-106
4.24.3	<b>Format</b> .....	4-106
4.24.4	<b>Description</b> .....	4-106
4.24.5	<b>Paramètre facultatif</b> .....	4-107
<b>4.25</b>	<b>QASSIGN</b> .....	4-108
4.25.1	<b>Fonction</b> .....	4-108
4.25.2	<b>Rubrique</b> .....	4-108
4.25.3	<b>Format</b> .....	4-108
4.25.4	<b>Description</b> .....	4-108
4.25.5	<b>Paramètres</b> .....	4-109
4.25.5.1	Paramètres obligatoires .....	4-109
4.25.5.2	Paramètres facultatifs .....	4-110
<b>4.26</b>	<b>RELEASE</b> .....	4-112
4.26.1	<b>Fonction</b> .....	4-112
4.26.2	<b>Rubrique</b> .....	4-112
4.26.3	<b>Format</b> .....	4-112
4.26.4	<b>Description</b> .....	4-112
4.26.5	<b>Paramètres</b> .....	4-113
4.26.5.1	Paramètre obligatoire .....	4-113
4.26.5.2	Paramètres facultatifs .....	4-113
<b>4.27</b>	<b>REPORT</b> .....	4-114
4.27.1	<b>Fonction</b> .....	4-114
4.27.2	<b>Rubrique</b> .....	4-114
4.27.3	<b>Format</b> .....	4-114
4.27.4	<b>Description</b> .....	4-114
<b>4.28</b>	<b>RUN</b> .....	4-115
4.28.1	<b>Fonction</b> .....	4-115
4.28.2	<b>Rubrique</b> .....	4-115
4.28.3	<b>Format</b> .....	4-115

## Table des matières

<b>4.28.4</b>	<b>Description</b> .....	4-116
<b>4.28.5</b>	<b>Paramètres</b> .....	4-117
4.28.5.1	Paramètres obligatoires .....	4-117
4.28.5.2	Paramètres facultatifs .....	4-117
<b>4.29</b>	<b>SEND</b> .....	4-121
<b>4.29.1</b>	<b>Fonction</b> .....	4-121
<b>4.29.2</b>	<b>Rubrique</b> .....	4-121
<b>4.29.3</b>	<b>Format</b> .....	4-121
<b>4.29.4</b>	<b>Description</b> .....	4-121
<b>4.29.5</b>	<b>Paramètres</b> .....	4-122
4.29.5.1	Paramètre obligatoire .....	4-122
4.29.5.2	Paramètres facultatifs .....	4-122
<b>4.30</b>	<b>SIZE</b> .....	4-123
<b>4.30.1</b>	<b>Fonction</b> .....	4-123
<b>4.30.2</b>	<b>Rubrique</b> .....	4-123
<b>4.30.3</b>	<b>Format</b> .....	4-123
<b>4.30.4</b>	<b>Description</b> .....	4-123
<b>4.30.5</b>	<b>Paramètres facultatifs</b> .....	4-124
<b>4.31</b>	<b>STEP</b> .....	4-126
<b>4.31.1</b>	<b>Fonction</b> .....	4-126
<b>4.31.2</b>	<b>Rubrique</b> .....	4-126
<b>4.31.3</b>	<b>Format</b> .....	4-126
<b>4.31.4</b>	<b>Description</b> .....	4-127
<b>4.31.5</b>	<b>Paramètres</b> .....	4-127
4.31.5.1	Paramètres obligatoires .....	4-127
4.31.5.2	Paramètres facultatifs .....	4-128
<b>4.32</b>	<b>\$SWINPUT</b> .....	4-131
<b>4.32.1</b>	<b>Fonction</b> .....	4-131
<b>4.32.2</b>	<b>Rubrique</b> .....	4-131
<b>4.32.3</b>	<b>Format</b> .....	4-131
<b>4.32.4</b>	<b>Description</b> .....	4-131
<b>4.32.5</b>	<b>Paramètres obligatoires</b> .....	4-132
<b>4.33</b>	<b>SYSOUT</b> .....	4-134
<b>4.33.1</b>	<b>Fonction</b> .....	4-134
<b>4.33.2</b>	<b>Rubrique</b> .....	4-134
<b>4.33.3</b>	<b>Format</b> .....	4-134
<b>4.33.4</b>	<b>Description</b> .....	4-135
<b>4.33.5</b>	<b>Paramètres</b> .....	4-137
4.33.5.1	Paramètre obligatoire .....	4-137
4.33.5.2	Paramètres facultatifs .....	4-137

<b>4.34</b>	<b>VALUES</b> .....	4-143
4.34.1	<b>Fonction</b> .....	4-143
4.34.2	<b>Rubrique</b> .....	4-143
4.34.3	<b>Format</b> .....	4-143
4.34.4	<b>Description</b> .....	4-143
4.34.5	<b>Paramètres facultatifs</b> .....	4-144
<b>4.35</b>	<b>WRITER</b> .....	4-146
4.35.1	<b>Fonction</b> .....	4-146
4.35.2	<b>Rubrique</b> .....	4-146
4.35.3	<b>Format</b> .....	4-146
4.35.4	<b>Description</b> .....	4-147
4.35.5	<b>Paramètres</b> .....	4-148
4.35.5.1	Paramètres obligatoires .....	4-148
4.35.5.2	Paramètres facultatifs .....	4-149
<b>5.</b>	<b>Ordres JCL étendus</b> .....	5-1

## Annexes

<b>A.</b>	<b>Liste des ordres JCL et de leurs abréviations .....</b>	A-1
<b>B.</b>	<b>Conversion des caractères.....</b>	B-1
<b>B.1</b>	<b>ASCII.....</b>	B-2
<b>B.2</b>	<b>EBCDIC.....</b>	B-3
	<b>Glossaire.....</b>	g-1
	<b>Index .....</b>	i-1

# Illustrations

## Figures

1-1	Rubrique de description de travail .....	1-2
1-2	Exemple de train d'entrées.....	1-4
1-3	Correspondance entre phases et états d'un travail.....	1-8
1-4	Rôle du lecteur de travaux et fichiers associés.....	1-10
1-5	Traitement d'un travail comportant une seule activité .....	1-19
3-1	Partie d'une structure arborescente .....	3-4
3-2	Règles d'unicité des noms .....	3-5

## Tableaux

1-1	Attributs implicites des classes de travaux (classes à un caractère) (1/2).....	1-15
1-1	Attributs implicites des classes de travaux (classes à un caractère) (2/2).....	1-16
3-1	Longueur maximum des noms de fichier .....	3-8
3-2	Valeur des paramètres .....	3-17
3-3	Syntaxe des noms figurant dans les ordres JCL.....	3-18
5-1	Récapitulatif des ordres JCL Etendus (1/4) .....	5-2
5-1	Récapitulatif des ordres JCL Etendus (2/4) .....	5-3
5-1	Récapitulatif des ordres JCL Etendus (3/4) .....	5-4
5-1	Récapitulatif des ordres JCL Etendus (4/4) .....	5-5
5-2	Explication des références bibliographiques du tableau 5-1 .....	5-6
A-1	Noms complets et abréviations des ordres JCL (1/3) .....	A-2
A-1	Noms complets et abréviations des ordres JCL (2/3) .....	A-3
A-1	Noms complets et abréviations des ordres JCL (3/3) .....	A-4

# 1. Introduction

Le présent chapitre décrit, de façon générale, la structure d'un travail utilisateur et le déroulement de son exécution sous GCOS 7. Il n'est donné ici qu'un bref aperçu du traitement des travaux utilisateur. Pour plus de détails et des exemples, se reporter au *Guide utilisateur JCL*.

## 1.1 DESCRIPTION D'UN TRAVAIL

### 1.1.1 Terminologie

Les programmes à exécuter sont normalement soumis au système d'exploitation sous la forme d'un **train d'entrées**, comportant une ou plusieurs **descriptions de travaux**.

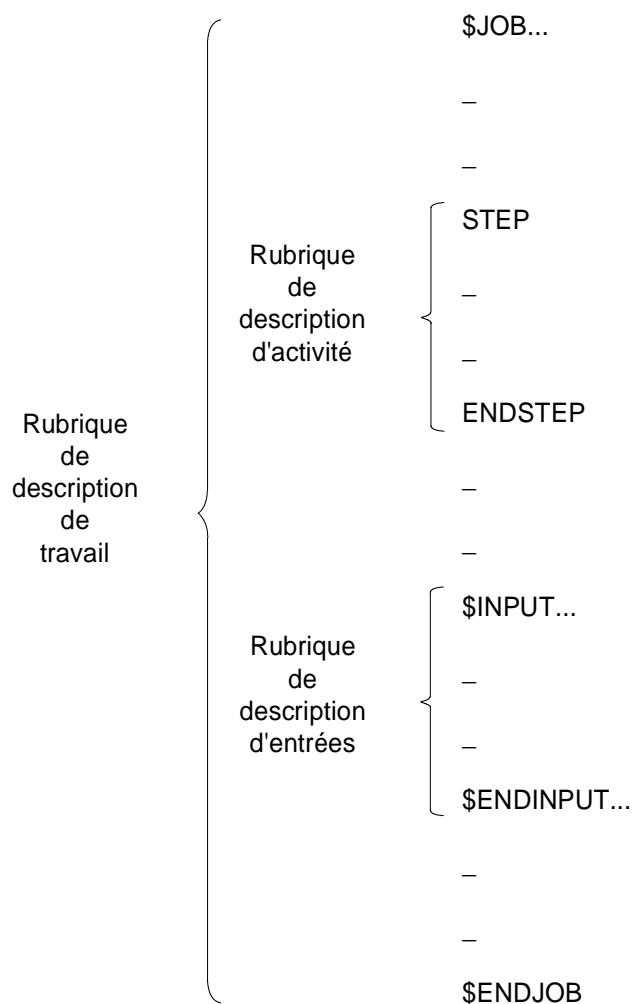
Chaque description de travail définit un travail à exécuter pour un utilisateur donné du système. Les descriptions de travaux se composent d'ordres de gestion des travaux ou **ordres JCL** (Job Control Language), qui fournissent des informations sur les travaux et sur la façon de les exécuter.

Le chargement des données d'entrée dans une bibliothèque peut se faire au moyen d'une **rubrique de description de données**, qui ne constitue pas un travail, mais se trouve au même niveau. La syntaxe des ordres JCL est présentée au chapitre 2.

### 1.1.2 Rubrique de description de travail

L'ensemble des ordres JCL constituant une **description de travail** forment une **rubrique de description de travail**.

Une rubrique de description de travail commence par un ordre \$JOB et se termine par un ordre \$ENDJOB. Elle comporte une ou plusieurs **rubriques de description d'activité** et éventuellement, une ou plusieurs **rubriques de description d'entrées** (voir figure 1-1).



**Figure 1-1. Rubrique de description de travail**

### 1.1.3 Rubrique de description d'activité

Une **activité** correspond au chargement et à l'exécution d'un programme qui, au moment de son exécution, constitue un **module chargeable**.

L'ensemble des ordres JCL qui définissent les caractéristiques et les besoins d'une activité forment une **rubrique de description d'activité**. Une description de travail peut comporter plusieurs rubriques de description d'activité. Comme le montre la figure 1-1, chaque rubrique de description d'activité commence par un ordre STEP et se termine par un ordre ENDSTEP.

### 1.1.4 Rubrique de description d'entrées

Une **rubrique de description d'entrées** commence par un ordre \$INPUT et se termine par un ordre \$ENDINPUT. Elle contient les articles de données à associer à une ou plusieurs activités du travail. Il peut s'agir, par exemple, de données à traiter par un module chargeable utilisateur ou d'un programme origine à compiler.

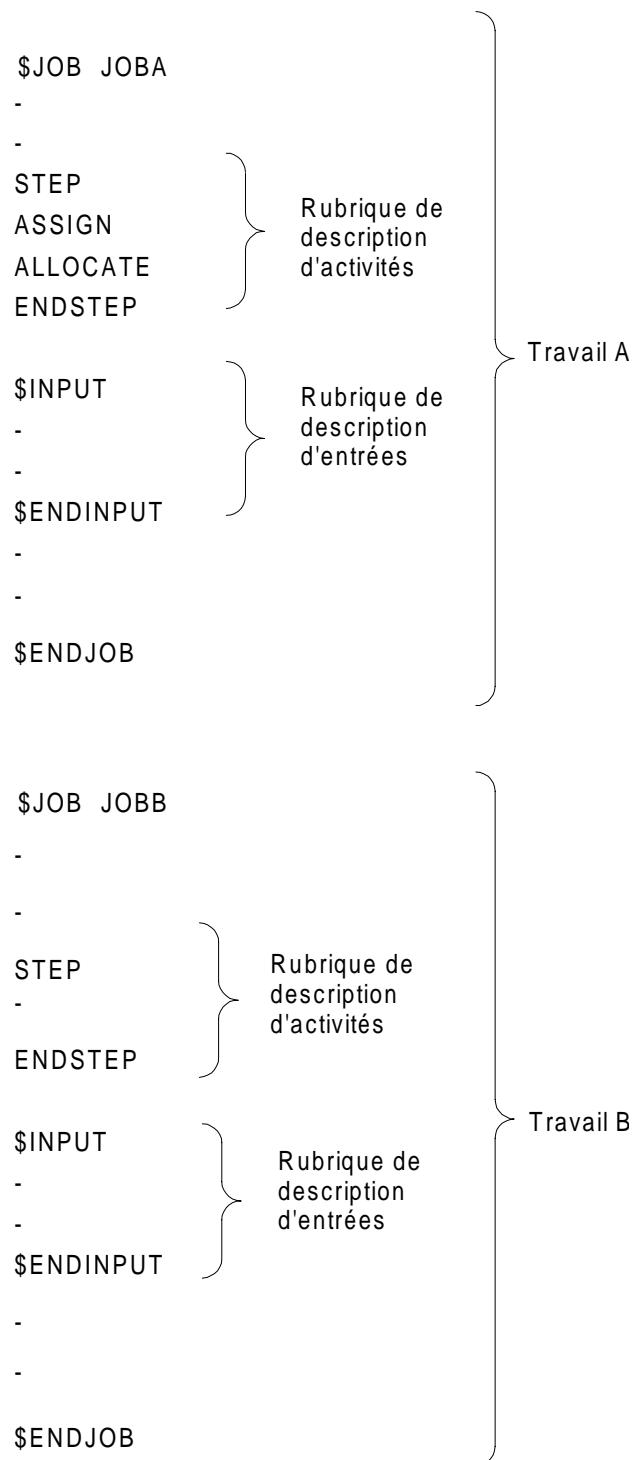
### 1.1.5 Rubrique de description de données

Une **rubrique de description de données** commence par un ordre \$DATA et finit par un ordre \$ENDDATA. Elle contient les données à charger dans la bibliothèque spécifiée. Cette rubrique se trouve au même niveau qu'une rubrique de description de travail.

### 1.1.6 Exemple de train d'entrées

La structure d'un train d'entrée comporte trois niveaux de descriptions qui jouent chacun un rôle particulier :

- La rubrique de description de travail qui déclare le travail au système : chaque travail d'un train d'entrée est délimité par un ordre \$JOBS et un ordre \$ENDJOB.
- La rubrique de description d'activité qui indique au système d'exploitation le traitement de l'activité : les besoins en ressources systèmes sont définis en ordres JCL encadrés à gauche par un ordre STEP et à droite par un ordre ENDSTEP.
- La rubrique de description d'entrées qui définit les entrées de données dans un train de données : les données spécifiées sont délimitées par des ordres \$INPUT et \$ENDINPUT.



**Figure 1-2. Exemple de train d'entrées**

## 1.2 TYPES D'ORDRES JCL

La plupart des ordres d'une description de travail sont des ordres de description d'activités. Autrement dit, ils désignent le module chargeable à exécuter et définissent les conditions et ressources nécessaires pour chaque activité.

Si une activité comprend l'exécution d'un programme système (comme un compilateur ou un utilitaire), il n'est pas nécessaire de fournir tous les ordres JCL qui la définissent : il suffit de spécifier un ordre JCL étendu (par exemple, COBOL ou LIBMAINT) accompagné des paramètres appropriés, celui-ci équivalant à un jeu d'ordres JCL de base. Au moment de la traduction de la description de travail en format interne, l'ordre unique spécifié par le programmeur est développé, c'est-à-dire remplacé par le jeu d'ordres JCL de base correspondants.

Le langage JCL comporte ainsi deux types d'ordres :

- les **ordres JCL de base**, qui demandent l'exécution d'une fonction donnée du système (par exemple STEP ou ASSIGN),
- les **ordres JCL étendus**, qui correspondent chacun à un jeu d'ordres JCL de base constituant une description complète d'activité (par exemple COBOL, SORT ou PREALLOC).

Le chapitre 4 décrit en détail les ordres JCL de base. Le chapitre 5 indique simplement la fonction des différents ordres JCL étendus et renvoie, pour plus de précisions, aux documents appropriés.

**Remarque :** Malgré leurs différences d'application, les deux types d'ordres JCL suivent les mêmes règles de syntaxe (voir chapitre 2).

Les points suivants sont à noter quant à la documentation et au traitement des deux types d'ordres JCL :

- Les ordres RUN et EXECUTE correspondent, développés, à une activité. Il ne s'agit cependant pas d'ordres utilitaires : ils sont en effet considérés comme faisant partie du système d'exploitation de base, et donc traités comme des ordres JCL de base (déscrits dans ce manuel).
- Tous les ordres JCL étendus, sauf SORTWORK et GSORTWRK, doivent figurer en dehors des rubriques de description d'activité.

## 1.3 SOUMISSION D'UN TRAVAIL

L'utilisateur peut soumettre les travaux au système d'exploitation :

- localement, sur votre site, à partir de fichiers magnétiques (disques, disquettes, bandes, etc.) ;
- à partir d'une station distante en traitement par lots (station RBF) et/ou en traitement distribué (station DJP).

**Remarque :** Un utilisateur IOF peut introduire certains ordres JCL directement (c'est-à-dire en dehors d'une rubrique de description d'activité). Pour plus de détails sur ce point, voir le manuel de référence de l'utilisateur IOF. Il peut en particulier soumettre au système un train de travaux enregistré, au moyen d'un ordre RUN.

## 1.4 PASSAGE D'UN TRAVAIL ET ETATS D'UN TRAVAIL

### Passage d'un travail

Le **passage d'un travail** est son cycle de vie complet, et non seulement la période au cours de laquelle il est exécuté.

Au cours de son passage, un travail est identifié de façon unique par **un numéro de passage** (RON - Run Occurrence Number). Il s'agit d'un simple numéro pouvant comporter jusqu'à quatre chiffres, et toujours précédé de la lettre X : par exemple, X12, X27, X118.

Les différentes activités d'un travail sont désignées par un **numéro d'activité**, qui correspond à leur ordre physique d'apparition dans la description du travail.

### Etats d'un travail

Le passage d'un travail peut être considéré comme une succession de phases, auxquelles correspondent différents **états du travail** (voir figure 1-3) :

- introduit (IN),
- en planification (SCH),
- bloqué (HOLD),
- en exécution (EX),
- suspendu (SUSP),
- en sortie (OUT).

Le travail est tout d'abord introduit (soumis à GCOS) ; autrement dit, une demande d'exécution du travail est envoyée à GCOS en même temps que la description du travail. Le travail devient ainsi "connu du système", et c'est à ce moment qu'un numéro de passage (RON) lui est attribué. La description du travail est ensuite traduite en format interne, adapté à l'exécution ; c'est au cours de l'opération de traduction que les erreurs de syntaxe JCL sont détectées.

Entre le moment où il est soumis à GCOS et la fin de la traduction du langage JCL, le travail est à l'état IN (introduit). Le **lecteur de travaux** sert à introduire et à traduire les descriptions de travaux.

Une fois traduit, le travail passe à l'état SCH (en planification). Le **planificateur de travaux** choisit (suivant certaines règles) le prochain travail à exécuter parmi tous les travaux se trouvant à l'état SCH. Le travail choisi passe alors à l'état EX (en exécution).

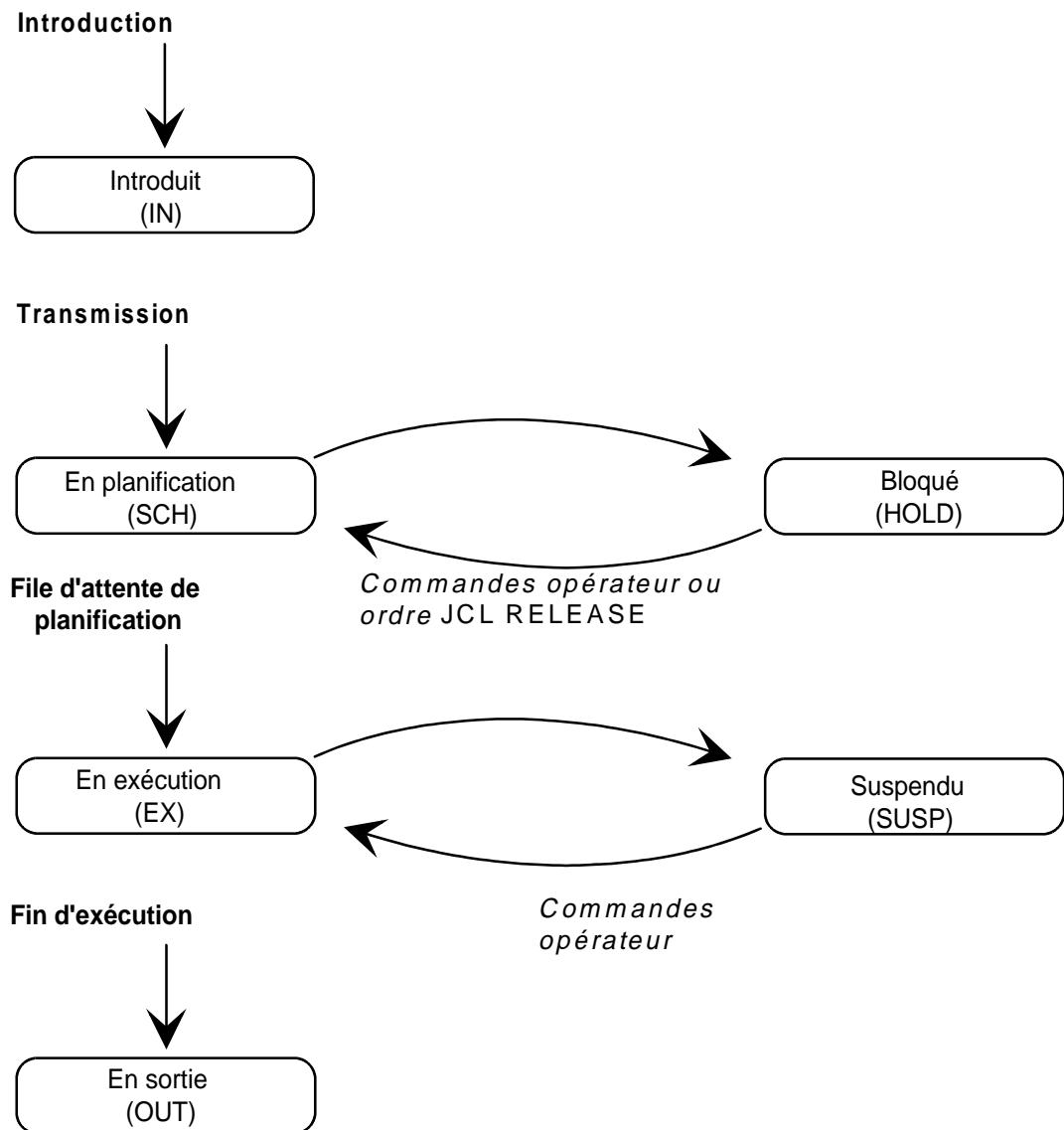
Une fois son exécution terminée, ce travail passe à l'état OUT (en sortie) et y demeure jusqu'à ce que l'**éditeur des sorties** ait imprimé tous ses résultats d'exécution, après quoi il cesse d'être connu du système.

Un travail peut se trouver dans deux autres états (moins fréquents) : l'état HOLD (bloqué) et l'état SUSP (suspendu). Les travaux bloqués se trouvent au même niveau que les travaux en planification, mais sont retirés de la file d'attente du planificateur de travaux et ne sont donc pas pris en considération par ce dernier. Un travail peut être placé à l'état HOLD par le paramètre HOLD de l'ordre JCL \$JOB, ou par la commande opérateur HOLD\_JOB (HJ).

Il est possible de faire passer un travail de l'état HOLD à l'état SCH au moyen de la commande opérateur RELEASE\_JOB (RJ) ou en spécifiant un ordre JCL RELEASE dans un autre travail. L'état HOLD permet de conditionner l'exécution d'un travail introduit à l'exécution d'un certain événement.

L'utilisateur peut employer la commande opérateur HOLD\_JOB (HJ) pour modifier l'ordre de planification des travaux. Cette commande peut également servir à interrompre momentanément les travaux en cours d'exécution, c'est-à-dire à les faire passer à l'état SUSP, possibilité parfois très utile pour régler des conflits de ressources répétés entre deux travaux.

L'utilisateur peut relancer l'exécution d'un travail suspendu au moyen de la commande opérateur RELEASE\_JOB (RJ). Dans ce cas, c'est la priorité d'exécution du travail (expliquée dans la suite du chapitre) qui détermine le moment où il est relancé.



**Figure 1-3. Correspondance entre phases et états d'un travail**

## 1.5 LECTEUR DE TRAVAUX

Le lancement du lecteur de travaux s'effectue au moyen d'une commande opérateur ou en chargeant une disquette spéciale.

Le lecteur de travaux lit un train de descriptions de travaux, analyse chaque travail, traduit le JCL en format interprétable par GCOS, et range séparément les données et les ordres JCL correspondants, en vue d'une planification et d'une exécution ultérieures du travail.

La traduction du JCL s'effectue en deux opérations successives : la lecture des entrées et la traduction proprement dite. Le lecteur des entrées et le traducteur JCL (JTRA) sont des modules chargeables distincts.

### 1.5.1 Lecteur des entrées

Lorsque le lecteur des entrées se voit demander de lire un train de travaux (voir figure 1-4), il affecte l'appareil d'entrée et attend le premier article. Les ordres JCL de chaque travail sont enregistrés en format origine dans un fichier mémoire de réserve, et chaque rubrique de description d'entrées est enregistrée séparément dans un fichier système appelé SYS.IN, ce qui donne, pour chaque travail, un fichier d'ordres JCL et autant de sous-fichiers SYSIN qu'il y a de rubriques de description d'entrées.

Lorsque le lecteur des entrées détecte un ordre \$SWINPUT, il passe du train de travaux en cours au fichier spécifié dans cet ordre. Lorsqu'il rencontre un ordre \$ENDJOB, il crée une rubrique dans une file d'attente à laquelle le traducteur JCL a accès. Le lecteur des entrées répète ces opérations pour le travail suivant, et ainsi de suite jusqu'à la fin du train de travaux. Il informe alors la fin des entrées au traducteur JCL et demande la traduction.

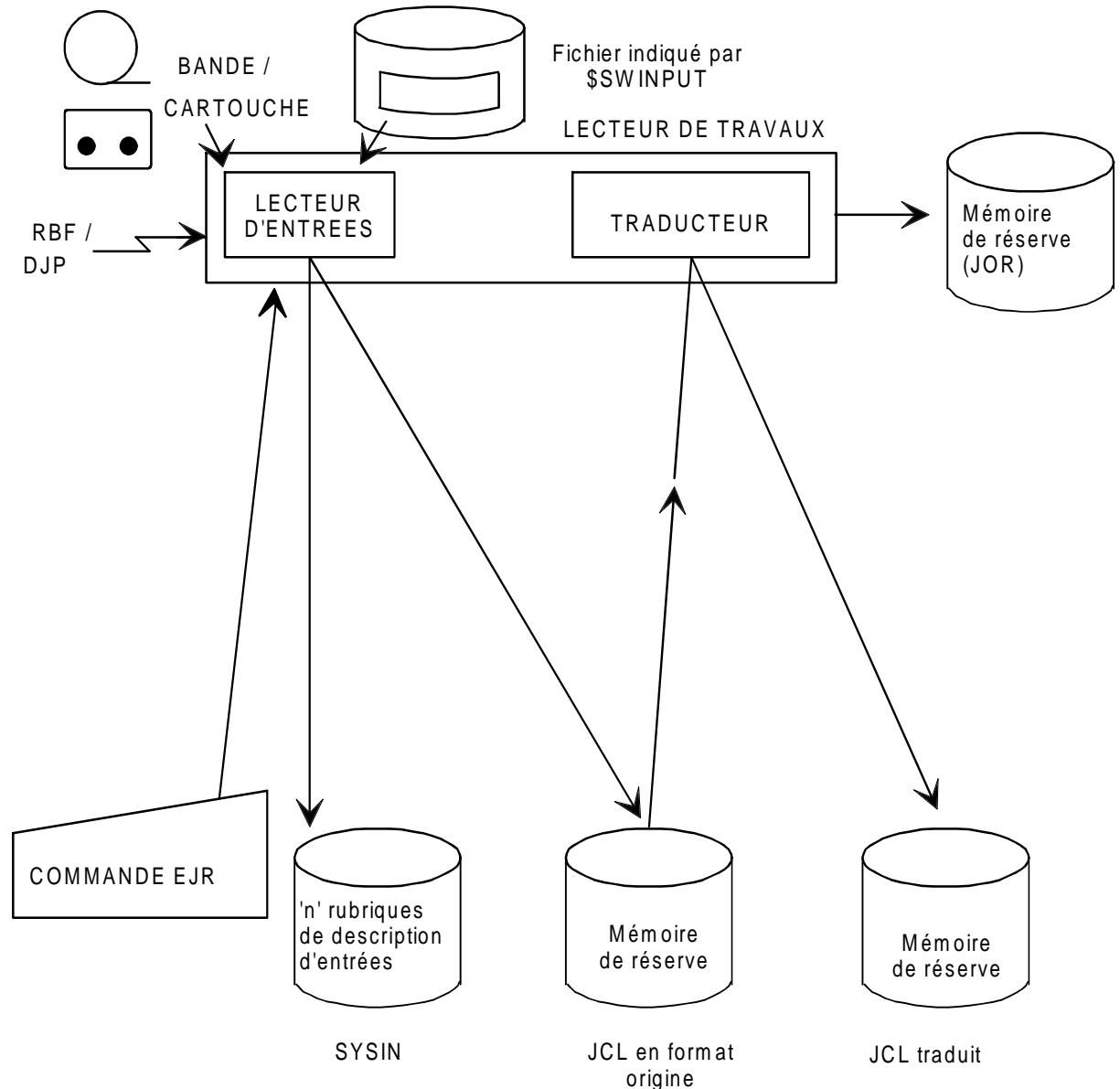
**Remarque :** Une fois les rubriques de description d'entrées séparées de la description du travail et les ordres JCL enregistrés dans le fichier d'ordres JCL par le lecteur des entrées, les ordres \$JOB et \$ENDJOB ne font plus partie de la description du travail et ne sont pas soumis au traducteur JCL.

Une fois les n rubriques de description d'entrées enregistrées dans le fichier SYS.IN, aucun ordre \$INPUT ou \$ENDINPUT ne fait plus partie de la description du travail.

De même, une fois que le lecteur des entrées a terminé la lecture du fichier spécifié dans l'ordre \$SWINPUT, cet ordre ne figure plus dans la description du travail.

### 1.5.2 Train d'entrées

La figure 1-4 ci-dessous illustre le rôle du lecteur de travaux.



**Figure 1-4. Rôle du lecteur de travaux et fichiers associés**

### 1.5.3 Traducteur JCL

Lorsque le lecteur des entrées signale au traducteur JCL la fin d'un train d'entrées, ce dernier commence à traduire les ordres JCL enregistrés en mémoire de réserve dans un format exécutable pour l'interpréteur d'ordres (voir paragraphe 1.8.2). Pour chacun des fichiers d'ordres JCL générés par le lecteur des entrées, le traducteur JCL ouvre un fichier mémoire de réserve dans lequel il écrit les ordres JCL traduits de la description de travail concernée. Il consigne également les erreurs JCL dans le JOR.

Bien que les travaux soient introduits par le lecteur de travaux dans l'ordre où ils figurent à l'intérieur du train d'entrées, ils sont sélectionnés pour traduction selon leur priorité de planification (indiquée dans l'ordre \$JOB).

## 1.6 PLANIFICATION DES TRAVAUX

Lorsque la traduction d'une description de travail est terminée, le planificateur de travaux détermine le moment où le travail peut être lancé (si le lecteur de travaux n'a trouvé aucune erreur).

### 1.6.1 Planificateur de travaux

Le planificateur de travaux gère la charge du système au niveau travail et offre à l'utilisateur diverses possibilités d'organisation et de planification en vue d'optimiser les performances du système. Il met également à la disposition de l'utilisateur une série d'ordres JCL et de commandes opérateur lui permettant d'intervenir sur le déroulement des travaux.

#### 1.6.1.1 File d'attente du planificateur de travaux

Une priorité de planification est associée à chaque travail, indiquant son urgence par rapport aux autres travaux. Les priorités de planification vont de 0 (priorité la plus élevée) à 7 (priorité la plus faible). Les travaux en planification (état SCH) intègrent la file d'attente du planificateur de travaux à la fin de la traduction du JCL, et la quittent lorsque leur exécution commence.

La priorité de planification d'un travail peut être spécifiée au moyen du paramètre PRIORITY de l'ordre \$JOB. Si elle n'est pas spécifiée, c'est la priorité associée à la classe dont il fait partie qui s'applique.

**Remarque :** L'utilisateur peut modifier les priorités de planification et d'exécution d'un travail, ainsi que son état.

### 1.6.2 Gestion des travaux

Le nombre de travaux concurremment exécutables est limité pour des raisons pratiques : en supposant par exemple que l'on dispose de trois dérouleurs de bande, il est inutile d'essayer d'exécuter deux travaux en même temps si chacun d'entre eux nécessite trois dérouleurs. GCOS 7 offre un ensemble de fonctions permettant de contrôler le nombre et le type de travaux exécutés concurremment, ainsi que l'ordre dans lequel ils sont exécutés.

Le nombre de travaux en exécution (état EX) et suspendus (état SUSP) à un moment donné constitue la **charge du système**. Un maximum général, appelé **niveau de multiprogrammation du système** ou **charge maximum du système**, limite le nombre de travaux possibles en ces deux états.

A tout moment, les travaux constituant la charge du système sont répartis en une ou plusieurs classes de travaux.

### 1.6.3 Classes de travaux

#### 1.6.3.1 Classes de travaux

Les classes de travaux sont un moyen pour l'utilisateur d'organiser sa charge de travail de façon à optimiser les performances du système en environnement de multiprogrammation.

***Exemples :***

- Certains travaux doivent être exécutés en série (l'un après l'autre), du fait de contraintes inhérentes à l'installation.
- Certains travaux doivent être exécutés à des périodes données (la nuit, etc.).
- Certaines combinaisons de travaux sont plus performantes que d'autres : l'exécution concurrente de travaux tributaires de l'unité centrale et de travaux tributaires des E/S est ainsi plus performante que l'exécution concurrente de travaux uniquement tributaires de l'unité centrale.
- Si plusieurs travaux utilisent les mêmes ressources (par exemple, dérouleurs de bande ou autres périphériques), ils ne doivent pas être planifiés pour le même moment, afin de réduire les délais d'attente.

Ainsi, tout travail utilisateur entrant dans le système appartient à une classe donnée, à laquelle l'utilisateur peut associer certaines caractéristiques appelées **attributs**, lui permettant de moduler les combinaisons de travaux et la planification de la charge. La ou les classes sous lesquelles un projet peut exécuter un travail sont spécifiées dans la description du projet, qui figure dans le catalogue de site. C'est l'administrateur système qui affecte les classes de travaux à un projet, au moyen des commandes MAINTAIN\_CATALOG "CRP" et "MDP".

Une classe de travaux est identifiée par une ou deux lettres comme suit :

A...Z, AA...AZ, BA...BZ, etc., jusqu'à ZA...ZZ

ce qui détermine une limite théorique de 702 classes. Le maximum de classes gérables par le planificateur de travaux est cependant de 256.

Les classes à un caractère (A à Z) sont initialisées à la configuration du système. Leurs attributs implicites sont résumés dans le tableau 1-1. Ces attributs peuvent être modifiés sous certaines conditions, mais la classe ne peut être supprimée.

Les classes à deux caractères doivent être créées puis initialisées au moyen de la commande START\_LOAD. Il est possible de les supprimer au moyen de la commande TERMINATE\_LOAD (voir le guide opérateur du système - 30UU pour la V3 et 60UU pour la V5).

- Les classes A à Q et AA à QZ sont réservées au travaux utilisateur.
- Les classes R à Z et RA à ZZ sont réservées aux travaux de service (elles ne peuvent pas être affectées aux travaux utilisateur).
- Les classes A à P et AA à PZ sont réservées aux travaux en traitement par lots, P étant leur classe implicite.
- Les classes A à Q et AA à QZ sont réservées aux travaux IOF, Q étant leur classe implicite.

Le nombre réel de travaux s'exécutant dans une classe à un moment donné constitue la **charge de la classe**. Il est possible de limiter le nombre de travaux s'exécutant concurremment dans cette classe en fixant un maximum, appelé **niveau de multiprogrammation de la classe** ou **charge maximum de la classe**.

Chaque classe de travaux a une priorité de planification implicite, des priorités implicites d'exécution des activités et une charge maximum implicite.

L'exécution d'une classe de travaux peut en outre être suspendue et relancée, ce qui constitue un moyen d'intervention supplémentaire sur l'ordre d'exécution des travaux.

#### 1.6.3.2 Attributs d'une classe de travaux

Un travail est planifié puis exécuté conformément à des règles associées à la classe dont il fait partie. Ces règles sont les attributs de la classe.

Chaque classe a trois attributs, qui servent à optimiser la charge de travail de l'utilisateur aussi longtemps que les travaux concernés existent. Ces attributs ne sont jamais supprimés par un arrêt normal ou un blocage du système et restent inchangés, sauf en cas de destruction de la mémoire de réserve permanente. Ils peuvent toutefois être modifiés par l'utilisateur, au moyen de la commande MODIFY\_JOB (MDJ).

Ces attributs sont les suivants :

##### **Priorité de planification**

La priorité de planification (PRIORITY) d'un travail détermine sa position dans la file d'attente de planification et le temps qu'il doit attendre pour être exécuté.

Les priorités de planification qu'il est possible d'attribuer à un travail peuvent être restreintes par la description du projet auquel ce travail est rattaché (catalogue de site). La priorité implicite retenue est celle du projet pour les travaux dont la priorité a été fixée sans tenir compte de cette restriction, ou pour ceux qui n'ont pas de priorité définie dans \$JOB ou RUN.

Si le projet n'a pas de priorité implicite, c'est la priorité implicite de la classe qui s'applique au travail.

La priorité de planification d'un travail peut être modifiée par l'utilisateur au moyen de la commande MODIFY\_JOB (MDJ).

### Priorité d'exécution

La priorité d'exécution d'un travail est déterminante en ce qui concerne l'obtention de mémoire et de temps CPU. C'est la valeur implicite appliquée à toutes les activités de tous les travaux d'une classe donnée pour lesquelles aucune priorité d'exécution (XPRTY) n'a été spécifiée dans l'ordre STEP.

La priorité d'exécution d'un travail peut être modifiée par l'utilisateur au moyen de la commande MODIFY\_JOB (MDJ).

### Charge maximum (niveau de multiprogrammation)

La charge maximum (MAXLOAD) d'une classe correspond au nombre maximum de travaux en exécution (EX) et suspendus (SUSP) dans cette classe.

Pour les classes A à Z, la valeur implicite des attributs est définie à la configuration du système. Le système est livré au client avec les classes de travaux et les attributs de classes implicites détaillés dans le tableau 1-1 ci-dessous.

Pour les classes à deux caractères, les valeurs implicites d'attributs sont fixées à la création de la classe.

**Table 1-1. Attributs implicites des classes de travaux (classes à un caractère) (1/2)**

Classe	Prorité de planification implicite (PRIORITY)	Priorité d'exécution implicite (XPRTY)	Charge maximale implicite (MAXLOAD)	Utilisation recommandée	Remarque
<b>TRAVAUX UTILISATEUR</b>					
A	7	9	1	Traitement par lots	
B	7	9	1	Traitement par lots	
C	7	9	1	Traitement par lots	
D	1	5	1	Traitement par lots	
E	2	4	1	Traitement par lots	
F	3	7	1	Traitement par lots	
G	4	9	1	Traitement par lots	
H	6	0	1	Communication	
I	7	9	4	Traitement par lots	
J	6	0	1	TDS	
K	7	9	1	Traitement par lots	
L	7	9	1	Traitement par lots	
M	7	9	1	Traitement par lots	
N	7	9	1	Traitement par lots	
O	7	9	1	Traitement par lots	
P	7	9	40	Traitement par lots (implicite)	

**Table 1-1. Attributs implicites des classes de travaux (classes à un caractère) (2/2)**

Classe	Prorité de planification implicite (PRIORITY)	Prorité d'exécution implicite (XPRTY)	Charge maximale implicite (MAXLOAD)	Utilisation recommandée	Remarque
<b>TRAVAUX DE SERVICE</b>					
Q	0	1	100	IOF	
R	0	1	6	Lecteurs	(1)
RA				OLTD (travail TD-MVH)	
RB				Miroir (travail REFFIL)	
S	0	0	12	Télécom (QMON, FECM, FEPS, TNS, RAEH, OCS) IOSER ARS	(1)
T	7	2	6	Télécom (travail SERVER2)	
U	0	1	6	RBF	
V	7	9	1	Suivi de système (travail TRCCL)	
W	0	1	8	Editeur de sortie (travail WRITER)	(1), (2)
X	0	2	4	Traducteur JCL (travail JTRA) Services TDS-HA (travaux CMSR, RECOV, JRU)	(1), (2)
Y	7	2	6	Télécom (travaux VCAM, GTP) Serveur de segment pour TDS (travail JPPC)	(1), (2)
Z	0	0	4	Opérateur principal (travail IOF) Gestionnaire de message local (travail LAEH)	(1)

(1) Classe toujours fixée

(2) Utiliser l'état IDLE

## 1.7 EXECUTION DES TRAVAUX

### 1.7.1 Sélection des travaux

Chaque fois que la charge réelle du système est inférieure à sa charge maximum, et que la file d'attente de planification contient au moins un travail appartenant à une classe dont la charge réelle est inférieure à sa charge maximum, un travail en file d'attente de planification est sélectionné pour exécution. La sélection s'effectue suivant les classes des travaux en exécution et en file d'attente.

### 1.7.2 Interpréteur d'ordres

Lorsque le planificateur de travaux sélectionne un travail en file d'attente, il en informe l'interpréteur d'ordres qui accède alors au langage JCL traduit correspondant, enregistré en mémoire de réserve. Pour chaque ordre JCL, l'interpréteur d'ordres lance la fonction correspondante du système d'exploitation.

Pour que l'exécution d'une activité puisse commencer, il faut que les demandes de ressources (zone tampon utilisateur, fichiers, volumes, appareils) que contient son JCL soient analysées. Si elles peuvent être satisfaites, l'activité est exécutée ; dans le cas contraire, elle est placée en attente.

### 1.7.3 Exécution d'une activité

A chaque ordre STEP, l'interpréteur d'ordres appelle la routine de lancement d'activité, qui lit tous les ordres de la rubrique de description d'activité et affecte les ressources nécessaires à l'activité.

Si l'affectation d'une ressource est impossible, l'activité doit attendre qu'elle devienne disponible. Une fois toutes les ressources nécessaires affectées à l'activité, le module chargeable approprié est chargé (à partir de la bibliothèque LM), et l'exécution de l'activité commence sur détection de l'ordre ENDSTEP.

### 1.7.4 Fin d'activité

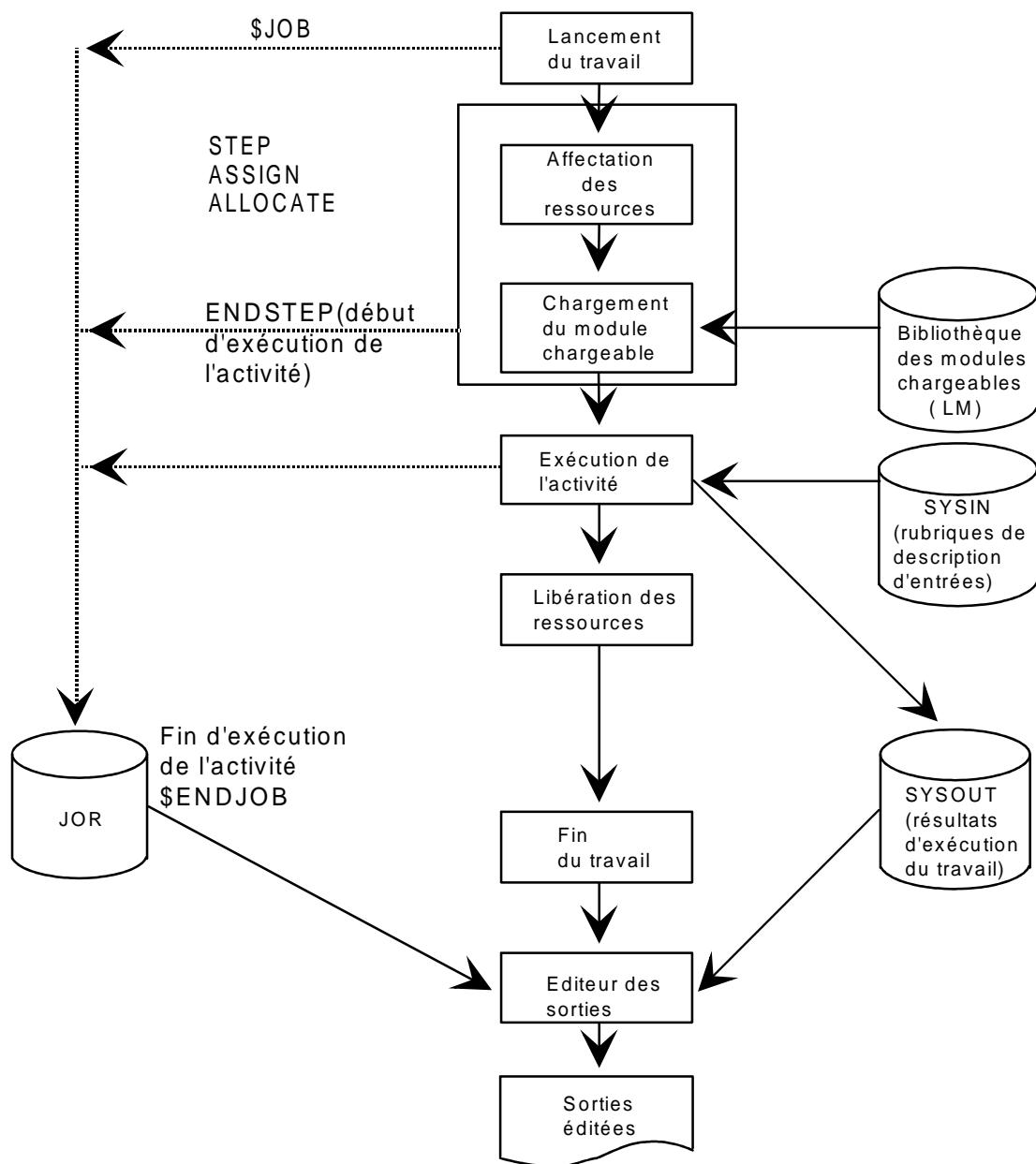
Lorsque l'exécution d'une activité est terminée, les ressources affectées sont libérées et l'interpréteur d'ordres reprend la main.

Si le travail comporte une autre activité, le cycle affectation des ressources / exécution de l'activité / libération des ressources se répète. A la fin du JCL traduit, l'exécution du travail est terminée. La figure 1-5 illustre le traitement d'un travail comportant une seule activité.

Certaines erreurs ou anomalies survenant au cours du lancement ou de l'exécution d'une activité peuvent provoquer son arrêt prématuré. Pour prévenir cette situation et éviter ainsi l'arrêt anormal de tout le travail concerné, il est possible d'utiliser l'ordre JUMP. L'ordre LET (avec le paramètre SEV) permet par ailleurs de forcer le code de gravité d'erreur d'une activité à une certaine valeur, afin de simuler une fin normale ou anormale de cette activité.

### 1.7.5 Fin de travail

Lorsque l'interpréteur d'ordres rencontre la fin d'un travail, certaines procédures sont lancées afin de supprimer les fichiers temporaires et les sous-fichiers SYSIN, générer des statistiques et informer l'opérateur. A ce stade, les résultats d'exécution du travail sont prêts pour l'éditeur des sorties.



**Figure 1-5. Traitement d'un travail comportant une seule activité**

## 1.8 TRAVAUX DE SERVICE

Un travail de service est un travail lancé par le système pour accomplir certaines tâches. Il n'est pas exécuté en continu, mais ponctuellement, pour fournir les fonctions nécessaires à un travail, ou activé temporairement pour effectuer certaines tâches.

Les travaux de services proposés sont les suivants :

- Moniteur interactif IOF
- Programme de lecture (READER)
- Tête de déplacement sur les disques T & D (TD\_MVH)
- Recharge automatique de disques miroirs (REFILL)
- Moniteur de files d'attente (QMON)
- Logiciel de gestion de frontal (FEPS)
- Logiciel d'administration de frontal (FECM)
- Sous-système de transport et réseau (TNS)
- Gestionnaire d'échanges administratifs à distance (RAEH)
- Sauvegarde et restauration automatiques (ARS)
- Sous-système ouvert de communications (OCS)
- Serveur d'E/S pour antémémoire LMC et/ou disques miroirs (IOSER)
- Serveur de transfert de fichiers DJP (SERVER2)
- Logiciel de traitement par lots (RBF)
- Suivi de système (TRCCL)
- Editeur des sorties (WRITER)
- Traducteur JCL (JTRA)
- Serveur de gestion de complexe HA (CMSR)
- Reconstitution de fichiers sur basculement HA (RECOV)
- Utilitaire de reconstitution de journal Après (JRU)
- Système de communication directe (VCAM)
- Processeur de transfert généralisé (GTP)
- Gestion de segments TDS (JPPC)
- Gestionnaire d'échanges administratifs local (LAEH)

**Remarques :**

1. Les travaux de service se distinguent des fonctions du système connues sous le nom de "processeurs système" (tels que les compilateurs ou les utilitaires de gestion de données et de bibliothèques) par le fait que les processeurs système sont appelés à partir des travaux utilisateur (par des ordres JCL étendus). Les ordres RUN et EXECUTE sont toutefois des exceptions, qui peuvent appeler des travaux de service à partir d'un travail utilisateur.
2. Un numéro de passage (RON) est attribué à tous les travaux de services lors de leur déclaration au système. La classe de travaux est exclusive à ce travail de service (voir plus haut le tableau 1-1).
3. Un seul travail JTRA peut être exécuté à la fois.
4. Un seul travail READER est lancé pour chaque appareil exécutant le programme de lecture.

5. Les travaux de services n'affichent pas de messages sur l'état d'avancement. Seuls des messages d'erreur sont communiqués.
6. Les travaux de service traversent les mêmes états que les travaux utilisateur, à cette différence près que certains travaux de services passent à l'état inactif (IDLE) lorsqu'ils cessent d'être utiles. Mais ils sont automatiquement remis en planification (SCH) dès qu'un travail se présente (transfert à effectuer, travail à traduire ou sortie à imprimer, par exemple). L'état IDLE permet d'optimiser l'utilisation des ressources système sans faire appel à l'opérateur.

## 1.9 FONCTIONS ACCESSIBLES AU MOYEN D'ORDRES JCL

Les paragraphes suivants indiquent les fonctions du système d'exploitation accessibles à l'utilisateur par l'intermédiaire d'ordres JCL de base. Pour plus de précisions, se reporter à l'ordre JCL voulu dans ce manuel ou au manuel utilisateur correspondant.

### 1.9.1 Désynchronisation des E/S sur périphériques autres que disques et bandes

La désynchronisation des entrées/sorties est une fonction du système donnant à l'utilisateur l'impression qu'un programme utilise directement un périphérique, alors qu'il passe en fait par un enregistrement intermédiaire dans des fichiers système sur disque, de façon à éliminer tout problème d'affectation d'appareils. Le périphérique équivaut en quelque sorte à un fichier séquentiel sur disque, à quelques restrictions de traitement près (il n'est par exemple pas possible de lire sur une imprimante). Pour plus de précisions sur ce point, voir le *Guide utilisateur des périphériques*.

## 1.10 GESTION DES ENTREES

### 1.10.1 Mécanisme SYSIN

Le mécanisme SYSIN permet l'exécution concurrente de programmes lisant des données sur un périphérique sans problèmes d'affectation des appareils. Le mécanisme SYSIN met en effet les fichiers d'entrée en mémoire tampon sur disque, ce qui accélère l'exécution des travaux. Certains travaux peuvent ainsi s'exécuter en même temps que sont lues les données destinées à un autre travail.

### 1.10.2 Utilisation du mécanisme SYSIN

Le mécanisme SYSIN utilise le fichier système SYS.IN, créé à la génération du système et enregistré sur un disque résident. Un sous-fichier de SYS.IN est affecté à chaque rubrique de description d'entrées. Le lecteur des entrées sépare chaque rubrique de description d'entrées de la description de travail correspondante et l'enregistre dans le sous-fichier de SYS.IN qui lui est affecté. Le format des données d'entrée est défini par le paramètre TYPE de l'ordre \$INPUT.

Ces données sont envoyées dans un sous-fichier du fichier SYS.IN. Pour accéder à ce sous-fichier, l'utilisateur appelle le mécanisme SYSIN et demande l'affectation du sous-fichier au moyen de l'ordre ASSIGN suivant :

```
ASSIGN nom-fichier-interne, *rubrique-de-description-d'entrées ;
```

Les sous-fichiers SYSIN sont temporaires : lorsque l'exécution d'un travail est terminée, le système supprime tous les sous-fichiers SYSIN qui lui étaient affectés.

## 1.11 GESTION DES SORTIES

### 1.11.1 Editeur des sorties

L'éditeur des sorties permet l'exécution concurrente de programmes envoyant des données à un périphérique de sortie (imprimante ou cassette) sans problèmes d'affectation des appareils. Il place en effet les fichiers d'impression ou de sortie en mémoire tampon externe, ce qui accélère l'exécution des travaux. Certains travaux peuvent ainsi s'exécuter en même temps que sont imprimés les résultats d'autres travaux dont l'exécution est terminée. Un fichier devant être traité par l'éditeur des sorties est appelé **fichier SYSOUT**. Il existe deux types de fichiers SYSOUT : les fichiers SYSOUT standard et les fichiers SYSOUT permanents.

### 1.11.2 Fichier SYSOUT standard (SYS.OUT)

Le fichier SYSOUT standard (appelé SYS.OUT) est un fichier système constitué à la génération du système et enregistré sur disque résident. Au cours d'une activité donnée, le système affecte un sous-fichier du fichier SYSOUT standard à chaque fichier de sortie généré par le programme qui est destiné au fichier SYSOUT standard. Le contenu des sous-fichiers SYSOUT standard est automatiquement imprimé par le système (implicitement en fin de travail). Lorsque le traitement des sorties est terminé, les sous-fichiers SYSOUT standard sont supprimés.

Un sous-fichier SYSOUT standard est créé lorsque les deux conditions suivantes sont réunies :

- L'utilisateur a déclenché le mécanisme SYSOUT (voir ci-après) dans le programme ou par un ordre JCL.
- Le fichier n'est pas affecté par un ordre JCL (ASSIGN). Le JOR est enregistré en mémoire de réserve et extrait par l'éditeur des sorties comme s'il s'agissait d'un sous-fichier SYSOUT.

### 1.11.3 Fichiers SYSOUT permanents

L'utilisateur demande un fichier SYSOUT permanent s'il veut :

- soit un fichier SYSOUT qui ne soit pas supprimé après traitement par l'éditeur des sorties,
- soit un fichier SYSOUT dont le contenu ne sorte qu'après l'activité en cours,
- soit un fichier SYSOUT sur bande (utile pour les sorties volumineuses).

Si l'utilisateur veut imprimer le contenu d'un fichier SYSOUT permanent après l'activité associée à la création de ce fichier, un ordre WRITER doit figurer dans la description de travail appropriée.

#### 1.11.4 Mécanisme SYSOUT et format du fichier SYSOUT

Le mécanisme SYSOUT est une façon d'écrire des données dans un fichier SYSOUT. L'utilisateur peut déclencher le mécanisme SYSOUT en plaçant le suffixe **-SYSOUT** après le nom de fichier interne approprié dans la clause **SELECT ... ASSIGN** d'un programme COBOL (**INPUT-OUTPUT SECTION**), ou au moyen de l'ordre JCL **SYSOUT**. Le fichier SYSOUT qui en résulte est dit "au format SYSOUT". Ce format intègre au fichier toutes les fonctions de mise en forme (définition des marges, densité d'impression, hauteur de feuillet, etc.) qui s'appliquent implicitement (spécifiées à la génération du système) ou que l'utilisateur a lui-même demandées (dans l'ordre **DEFINE**, par exemple). Un fichier au format SYSOUT est appelé **fichier édité**. L'utilisation du mécanisme SYSOUT pour générer des fichiers SYSOUT édités au moment de l'exécution d'un travail implique un traitement ultérieur moins important de la part de l'éditeur des sorties, lorsque le fichier est envoyé au périphérique de sortie.

S'il n'existe pas d'affectation pour un fichier à écrire en format SYSOUT, le système affecte automatiquement un sous-fichier SYSOUT standard. Pour qu'un fichier SYSOUT permanent soit créé, l'utilisateur doit affecter ce fichier (par un ordre **ASSIGN**) à l'activité au cours de laquelle le fichier est écrit.

Le mécanisme SYSOUT s'utilise toujours pour écrire dans le fichier SYSOUT standard. L'utilisation ou la non-utilisation du mécanisme SYSOUT pour la création d'un fichier SYSOUT permanent dépend de la longueur et du format d'article déclarés pour le fichier (pour plus de précisions sur ce point, voir la description de l'ordre **SYSOUT** au chapitre 3 et le *Guide utilisateur JCL*).

## 1.12 JCL ENREGISTRÉ

Une ou plusieurs descriptions de travaux peuvent être enregistrées sous la forme d'un train de travaux dans une bibliothèque origine. Ce train de travaux enregistré peut être tenu à jour comme n'importe quelle autre unité de bibliothèque origine, au moyen de LIBMAINT SL (voir le *Guide utilisateur* et le *Manuel de référence* traitant de la *maintenance des bibliothèques* ).

Un train de travaux enregistré peut être introduit dans le système au moyen de l'ordre JCL RUN ou de la commande opérateur EJR. L'ordre JCL \$SWINPUT permet d'insérer des ordres JCL enregistrés et/ou des articles de données dans une rubrique de description de travail ; cette opération s'effectue à l'exécution du lecteur des entrées.

L'ordre INVOKE permet à l'utilisateur d'introduire une séquence enregistrée de rubriques de description d'activités, d'ordres JCL étendus ou les deux dans une description de travail, au moment de la traduction du JCL ; cette séquence dite **séquence appelée** vient remplacer l'ordre INVOKE dans la description du travail.

L'ordre EXECUTE permet à l'utilisateur, pendant l'exécution d'un travail, de traduire et d'exécuter une séquence enregistrée de rubriques de description d'activités, d'ordres JCL étendus, ou des deux.

Pour plus de précisions sur ces points, voir le *guide utilisateur JCL*.

### 1.12.1 Mini-éditeur

Le mini-éditeur permet à l'utilisateur de modifier, au moment de la traduction du JCL, une séquence d'ordres JCL enregistrée ou faisant partie d'une rubrique de description d'entrées.

L'utilisateur dispose à cet effet d'une série de commandes de mise en forme spéciales. Les commandes nécessaires à une mise à jour donnée doivent être enregistrées dans une rubrique de description d'entrées. Le paramètre UPDATE de l'ordre INVOKE désigne la rubrique de description d'entrées contenant les commandes et lance le mini-éditeur pour mettre à jour le contenu de la séquence appelée.

Les règles d'utilisation du mini-éditeur sont exposées dans le *Guide utilisateur JCL*.

### 1.12.2 Références aux paramètres dans les ordres JCL

Une même description de travail ou une même séquence enregistrée d'ordres JCL peut se retrouver de multiples fois, à quelques valeurs de paramètres près (noms de fichiers, noms de volumes, etc.). L'utilisation des références aux paramètres permet d'éviter n répétitions de la même description/séquence en ne changeant à chaque fois que quelques valeurs de paramètres.

Le principe consiste à remplacer, dans la description/la séquence, les valeurs de paramètres par de simples références.

A la traduction, JTRA remplace toutes les références aux paramètres par les valeurs réelles correspondantes.

Ces valeurs sont définies dans les ordres JCL VALUES, MODVL, RUN, INVOKE, EXECUTE, et dans la commande opérateur EJR.

## 1.13 PARTAGE ET CONSERVATION DE FICHIERS

A l'intérieur d'une activité, il est possible d'accéder plusieurs fois au même fichier physique (identifié par un nom de fichier externe) au moyen de plusieurs fichiers logiques (identifiés par des noms de fichiers internes) ; les accès étant successifs, il n'existe pas de problèmes de concurrence.

En revanche, lorsque plusieurs travaux s'exécutant simultanément veulent accéder au même fichier, il peut en résulter des problèmes de concurrence et des modifications du fichier incompatibles avec les exigences d'un travail ; d'où la nécessité de réglementer l'accès au fichier et le partage du fichier, pour en assurer l'intégrité : tel est le rôle des paramètres SHARE et ACCESS de l'ordre ASSIGN.

Le partage de fichiers est expliqué plus en détail dans le *Guide utilisateur JCL*.

Le contenu d'un fichier temporaire peut être conservé d'une activité à une autre en spécifiant dans l'ordre ASSIGN soit END=PASS, soit en cas d'arrêt anormal d'activité, ABEND=PASS. Plusieurs activités d'un travail peuvent donc accéder au même fichier temporaire.

Pour un fichier permanent, la conservation d'une activité à une autre signifie que l'accès au fichier peut être réservé de l'une à l'autre ; tant que le fichier est conservé, aucun autre travail ne peut l'affecter.

Des exemples de conservation de fichiers sont donnés dans le *Guide utilisateur JCL*.

## 1.14 UTILISATION DES COMMANDES OPÉRATEUR

L'utilisateur peut incorporer des commandes opérateur à une description de travail (en fonction des restrictions imposées par le projet auquel est rattaché le travail). L'ordre JCL EXDIR est spécialement prévu à cet effet.

## 1.15 RELANCE SUR POINT DE REPRISE

Il peut arriver, pour diverses raisons (blocage du système, panne de courant, incident périphérique ou arrêt prématuré d'un programme suite à une erreur de l'opérateur ou à une erreur d'E/S), que le traitement d'un programme utilisateur se termine de façon anormale. La constitution de points de reprise et la relance sur point de reprise permettent de récupérer les données perdues.

C'est le paramètre REPEAT de l'ordre STEP ou de l'ordre \$JOB qui spécifie le caractère relançable ou non de l'activité ou du travail (appel à la fonction point de reprise).

Pour plus de précisions concernant la relance sur point de reprise, se reporter au *Guide utilisateur JCL*, au *Manuel de l'administrateur du système* et au *Guide utilisateur de GAC-Extended*.

## 1.16 JOURNALISATION

Le mécanisme de relance sur point de reprise est complété par celui de la journalisation :

Le **journal Avant** sert à remplacer les fichiers dont la mise à jour a été interrompue par l'arrêt anormal du programme, dans l'état où ils se trouvaient au dernier point de reprise exécuté. Il s'agit donc d'une **restauration non actualisée**. Ce journal est requis si le repositionnement du fichier ne suffit pas à le remettre dans l'état antérieur à l'interruption.

Le **journal Après** sert à remplacer ces fichiers dans l'état exact où ils se trouvaient au moment de l'incident. Il s'agit donc d'une **restauration actualisée**.

Les journaux à utiliser sont spécifiés soit dans le catalogue, soit dans l'ordre DEFINE : paramètre BEFORE (journal Avant), AFTER (journal Après), BOTH (journal Avant et Après) ou NONE (aucun journal).

L'ordre DEFINE est associé aux fichiers concernés au moyen des noms de fichiers internes. L'activité correspondante peut ou non utiliser le mécanisme de relance sur point de reprise.

## 2. Règles d'écriture des ordres JCL

Un ordre JCL peut comporter les éléments suivants :

- une étiquette
- un nom d'ordre
- un ou plusieurs paramètres.

Le format d'un ordre JCL est le suivant :

```
[étiquette :]...[$]nom-d'ordre [paramètre-positionnel]  
[paramètre-positionnel]...[paramètre-à-mot-clé]...  
[paramètre-autodéfini]...;
```

### 2.1 GENERALITES

1. Le symbole dollar (\$) est obligatoire uniquement pour les ordres traités par le lecteur des entrées (\$JOB et \$ENDJOB, \$INPUT et \$ENDINPUT, \$SWINPUT, \$DATA et \$ENDDATA), et pour les ordres JCL indiquant la fin d'une rubrique de description d'entrées pour laquelle END=DOLLAR a été spécifié (voir \$INPUT).

**Remarques :** Le symbole dollar peut être omis dans le premier ordre \$JOB d'un train de travaux enregistré (dont l'exécution est lancée par l'ordre RUN ou la commande EJR). Il est toutefois recommandé de ne jamais omettre le symbole dollar de l'ordre \$JOB.

Dans la documentation GCOS 7, le symbole \$ n'a pas été indiqué devant les ordres JCL (sauf pour \$JOB et \$ENDJOB, \$INPUT et \$ENDINPUT, \$SWINPUT, \$DATA et \$ENDDATA).

2. Le nom de l'ordre doit suivre immédiatement le symbole \$.
3. Le point-virgule (;) est le caractère fin obligatoire de tout ordre JCL.
4. La plupart des ordres JCL comportent des paramètres entre le nom de l'ordre et le caractère fin.

Il faut un espace au moins entre le nom de l'ordre et le premier paramètre.

Chaque paramètre doit être séparé du suivant (s'il en existe un) soit par une virgule, soit par un ou plusieurs espaces.

Lorsqu'une virgule sert de séparateur, elle peut être précédée ou suivie d'un ou plusieurs espaces.

**Remarque :** Dans ce manuel, le caractère choisi comme séparateur pour les formats d'ordres est l'espace. Pour les exemples avec deux paramètres ou plus sur une même ligne, la virgule peut néanmoins être utilisée comme séparateur, afin de faciliter la lecture.

5. N'importe quel ordre JCL peut figurer sur la même ligne qu'un autre ordre JCL ou plus, sauf les ordres traités par le lecteur des entrées (\$JOB et \$ENDJOB, \$INPUT et \$ENDINPUT, \$SWINPUT, \$DATA et \$ENDDATA), qui peuvent occuper chacun plus d'une ligne, mais ne doivent pas figurer sur la même ligne qu'un autre ordre du lecteur des entrées, un ordre JCL ou une rubrique de description d'entrées.
6. Les ordres obligatoirement précédés du symbole dollar ne doivent pas comporter d'étiquette (bien que \$ENDJOB puisse avoir une étiquette, si celle-ci figure sur la ligne précédente).
7. Lorsque le symbole dollar est obligatoire, il doit être le premier caractère de la ligne.
8. Un ordre pour lequel le symbole dollar n'est pas obligatoire (même s'il est mentionné) peut commencer à n'importe quelle position d'une ligne et se terminer à n'importe quelle position de cette ligne ou de la ligne suivante.

**Remarque :** Afin d'améliorer la lisibilité des descriptions de travail, il est conseillé de décaler les ordres JCL en fonction de leur niveau.

9. Tout paramètre, mot-clé ou valeur d'un ordre JCL peut commencer sur une nouvelle ligne.

Le système reconnaît la fin d'un ordre au point-virgule.

Dans une description de travail, la fin de ligne n'est pas considérée comme un séparateur.

10. Lorsqu'un ordre JCL occupe plus d'une ligne, un mot-clé ou une valeur de paramètre peut se trouver à cheval sur deux lignes, mais à condition :
  - que le dernier caractère de la première partie du nom ou de la valeur figure à la dernière position de caractère de la ligne ;
  - que le premier caractère de la seconde partie du nom ou de la valeur figure à la première position de caractère de la seconde ligne.

**Remarque :** Si la valeur de paramètre est une chaîne protégée, elle peut aussi figurer sur deux lignes, à condition d'utiliser un trait d'union (voir le paragraphe 2.2.3.3).

11. Les références aux paramètres ne sont pas autorisées dans les ordres pour lesquels le symbole dollar est obligatoire (\$JOB et \$ENDJOB, \$INPUT et \$ENDINPUT, \$SWINPUT, \$DATA et \$ENDDATA).

## 2.2 ELEMENTS D'UN ORDRE JCL

Un ordre JCL est composé des éléments suivants :

- étiquette,
- nom d'ordre,
- paramètres.

### 2.2.1 Etiquette

Tout ordre JCL (à l'exception de \$JOB, \$ENDJOB, \$INPUT, \$ENDINPUT, \$SWINPUT, \$DATA, \$ENDDATA) peut avoir une ou plusieurs étiquettes.

Une étiquette peut comporter jusqu'à huit caractères parmi les lettres de A à Z, les chiffres de 0 à 9, le trait d'union et le caractère de soulignement.

Elle doit être suivie d'un deux-points (:).

Les deux-points peuvent être précédés ou suivis d'un espace ou plus.

**Remarque :** L'étiquette CONTINUE a un rôle particulier, et n'est autorisée que dans les conditions expliquées pour l'ordre JUMP (voir chapitre 4). L'ordre \$ENDJOB peut comporter une étiquette, mais celle-ci doit figurer sur la ligne précédente. Par exemple :

FINISH:  
\$ENDJOB ;

est correct, mais

FINISH: \$ENDJOB ;

ne l'est pas.

### 2.2.2 Nom d'ordre

Le nom de l'ordre indique l'opération effectuée par le système. Le symbole dollar est nécessaire uniquement dans les cas indiqués au point 1 du paragraphe 2.1.

Le nom de l'ordre doit être écrit tel qu'il figure dans le format (chapitre 4). Il doit être suivi d'au moins un espace, sauf en l'absence de paramètres, auquel cas il est immédiatement suivi d'un point-virgule. Lorsqu'il s'agit d'un ordre obligatoirement précédé du symbole dollar, ce symbole doit figurer à la première position de caractère de la ligne contenant le nom de l'ordre.

**Exemples de noms d'ordres :**

- |        |   |
|--------|---|
| \$JOB  | Identifie un travail.   |
| STEP   | Désigne un module chargeable (pour l'exécution d'une activité). |
| ASSIGN | Affecte un fichier à une activité.                              |

## 2.2.3 Paramètres

Les paramètres commandent la façon dont le système interprète l'ordre où ils apparaissent. Certains ordres comme ENDSTEP n'ont pas de paramètres, mais il s'agit d'une exception et non du cas général.

### 2.2.3.1 Types de paramètres

Il existe trois types de paramètres :

- les paramètres positionnels,
- les paramètres à mots-clés,
- et les paramètres autodéfinis.

Les paramètres positionnels (s'il y en a) doivent être les premiers paramètres suivant le ou les espaces après le nom de l'ordre.

Si une virgule sert de séparateur entre deux paramètres, elle peut être précédée ou suivie d'un ou plusieurs espaces.

Certains ordres peuvent comporter un groupe de paramètres entre parenthèses à la place d'un seul paramètre. Les paramètres d'un groupe doivent être séparés par des virgules ou des espaces, et les paramètres positionnels doivent être les premiers paramètres à la suite de la parenthèse d'ouverture.

#### Paramètres positionnels

Le traducteur JCL reconnaît un paramètre positionnel à la position qu'il occupe dans l'ordre JCL. Les paramètres positionnels figurent juste après le ou les espaces qui suivent le nom de l'ordre (ou juste après la parenthèse d'ouverture d'un groupe de paramètres).

#### *Exemple :*

```
ASSIGN INFILE, PAYROLL.MAST;
```

Cet ordre comporte deux paramètres : INFILE est un nom de fichier interne et PAYROLL.MAST un nom de fichier externe.

Il s'agit de paramètres positionnels de l'ordre ASSIGN, qu'il faut donc impérativement faire figurer à la position indiquée, car le traducteur JCL les reconnaît à leur position.

Si un paramètre positionnel est omis dans un ordre ou dans un groupe de paramètres, deux virgules doivent séparer le paramètre précédent et le paramètre suivant (une seule virgule, s'il s'agit du premier paramètre positionnel d'un ordre ou d'un groupe de paramètres).

Si l'espace est utilisé comme séparateur, c'est le symbole "#" qui représentera le paramètre omis.

***Exemples :***

VALUES, P2,, P4;

équivaut à VALUES

# P2 # P4;

P2 est le deuxième paramètre positionnel et P4 le quatrième. Le premier et le troisième sont omis.

VALUES P2, P4;

équivaut à

VALUES P2 P4;

P2 est le premier paramètre positionnel et P4 le second.

Si tous les paramètres positionnels faisant suite à un paramètre positionnel donné sont omis, les virgules ou les symboles # correspondants peuvent également être omis.

***Exemples :***

VALUES P1, P2,,P4,,,; ou  
VALUES P1 P2 # P4 # #;

peuvent également s'écrire :

VALUES P1, P2,,P4; ou  
VALUES P1 P2 # P4;

Un paramètre positionnel peut être une valeur isolée (par exemple, un nom de fichier externe), une chaîne protégée (voir paragraphe 2.2.3.3) ou un groupe de paramètres.

Si seul le premier paramètre positionnel d'un groupe de paramètres est nécessaire, les parenthèses peuvent être omises, à condition que le paramètre positionnel ne soit pas lui aussi un groupe de paramètres.

***Exemple :***

INVOKE BJC, (BIG.YIN);

peut également s'écrire :

INVOKE BJC, BIG.YIN;

Les paramètres à mot-clé et les paramètres autodéfinis (voir pages suivantes) doivent toujours figurer après les paramètres positionnels (s'il y en a).

### Paramètres à mot-clé

Un paramètre à mot-clé se reconnaît à son nom. La position des paramètres à mot-clé par rapport aux autres paramètres d'un ordre est donc sans importance (à condition qu'ils viennent toujours après les paramètres positionnels). Les paramètres à mot-clé peuvent figurer dans l'ordre qui convient au programmeur.

Un paramètre à mot-clé peut se présenter sous l'une des formes suivantes :

- mot-clé=valeur,
- mot-clé='chaîne protégée',
- mot-clé=(groupe-de-paramètres),

Un mot-clé doit être écrit exactement tel qu'il apparaît dans le format de l'ordre (voir chapitre 4).

Le signe égal (=) peut être précédé ou suivi d'un ou plusieurs espaces. Comme indiqué pour les paramètres positionnels, un paramètre à mot-clé peut être une valeur isolée, une chaîne protégée ou un groupe de paramètres. Si seul le premier paramètre positionnel d'un groupe est nécessaire, les parenthèses peuvent être omises à condition que ce paramètre ne soit pas lui aussi un groupe de paramètres.

#### *Exemples :*

```
ASSIGN MFIL1, PAYROLL.MAST, DEVCLASS=MS/M500, MEDIA=VOLA1;
$INPUT MYFILE, TYPE=COBOL, END=ENDINPUT;
```

Dans les ordres ci-dessus,

- les paramètres positionnels sont MFIL1, PAYROL.MAST et MYFILE ;
- les noms de paramètres à mot-clé sont DEVCLASS, MEDIA, TYPE et END ;
- les valeurs de paramètres à mot-clé sont MS/M500, VOLA1, COBOL et ENDINPUT.

### Paramètres autodéfinis

Un paramètre autodéfini se reconnaît à son nom. Sa position dans l'ordre est donc sans importance, à condition que les paramètres positionnels (s'il y en a) figurent en premier.

Ce qui le distingue d'un paramètre à mot-clé est qu'il n'a pas de valeur associée : le nom lui-même du paramètre indique au système l'opération à effectuer. Un paramètre autodéfini doit être écrit exactement tel qu'il apparaît dans le format de l'ordre (voir chapitre 4).

#### *Exemple :*

```
$JOB JB2A, USER=XACC, HOLD;
```

Dans cet ordre, JB2A est une valeur de paramètre positionnel, USER un paramètre à mot-clé de valeur XACC, et HOLD un paramètre autodéfini.

**Remarque :** Le système ne distingue généralement pas un paramètre autodéfini d'un paramètre positionnel s'il figure à la même place : son nom est alors pris comme valeur du paramètre positionnel. L'utilisateur peut éviter cette confusion en plaçant, avant le premier paramètre autodéfini, soit un paramètre à mot-clé, soit autant de virgules ou de symboles # qu'il manque de paramètres positionnels.

**Exemples :**

Les ordres suivants ne sont pas équivalents :

```
ASSIGN IFN1, TEMPRY, RESIDENT;  
ASSIGN IFN1, RESIDENT, TEMPRY;
```

De même pour les deux ordres suivants :

```
ASSIGN FIL1, DEFER, DEVCLASS=MT/T9, MEDIA=VOLA;  
ASSIGN, FIL1, DEFER, DEVCLASS=MT/T9, MEDIA=VOLA;
```

Le dernier ordre ci-dessus peut également s'écrire :

```
ASSIGN # FIL1, DEFER, DEVCLASS=MT/T9, MEDIA=VOLA;
```

#### 2.2.3.2 Valeurs implicites des paramètres

Il est possible, dans la plupart des cas, d'omettre un ou plusieurs paramètres dans un ordre JCL donné.

Les paramètres positionnels sont presque tous obligatoires, alors que les paramètres à mot-clé ou les paramètres autodéfinis peuvent souvent ne pas être mentionnés. En effet, le système affecte automatiquement une valeur implicite à chaque paramètre omis : ainsi, dans l'ordre \$INPUT, les paramètres à mot-clé TYPE et END ont respectivement DATA et ENDINPUT pour valeur implicite.

Les ordres suivants sont donc équivalents :

```
$INPUT MYFILE;  
$INPUT MYFILE, TYPE=DATA, END=ENDINPUT;
```

#### 2.2.3.3 Chaînes protégées

Une chaîne protégée est une suite de caractères entre apostrophes.

Le contenu de la chaîne est protégé en ce sens qu'il n'est analysé ni par le lecteur des entrées, ni par le traducteur JCL. Une chaîne protégée peut donc comporter n'importe quel caractère du jeu en cours (espace compris). Les apostrophes sont considérées comme ne faisant pas partie de la chaîne protégée.

***Exemples :***

'A+B'

'A B'

'SMITH; '

Si une apostrophe doit apparaître à l'intérieur d'une chaîne protégée, il faut la doubler.

***Exemple :***

La chaîne

ABC'123+;

doit être écrite :

'ABC''123+;'

Une chaîne protégée peut s'étendre sur plusieurs lignes (si, par exemple, elle est trop longue pour tenir sur la seule ligne en cours). La coupure peut se faire à n'importe quel endroit de la ligne, c'est-à-dire pas obligatoirement à la dernière position de caractère, mais :

- le dernier caractère de la première partie de la chaîne doit être suivi d'un trait d'union et le reste de la ligne doit être entièrement constitué d'espaces ;
- le premier caractère de la seconde partie de la chaîne doit occuper la première position de caractère de la ligne suivante.

***Exemple :***

```
REPORT 'C"EST UN-
EXEMPLE DE CHAINE PROTEGEE LONGUE' ;
```

## 3. Identification des fichiers et groupes de paramètres standard

Ce chapitre comprend deux parties :

- La première partie expose les règles d'identification des fichiers, ainsi que les règles d'identification des volumes et appareils sur lesquels sont enregistrés les fichiers. Ces règles s'appliquent à tous les fichiers mentionnés dans les ordres JCL, même s'il est surtout fait état de l'ordre JCL de base ASSIGN, qui associe le nom d'un fichier dans un programme au nom sous lequel il est connu du système.
- La seconde partie détaille les formats des groupes de paramètres qui apparaissent dans plusieurs ordres JCL étendus.

### 3.1 IDENTIFICATION DES FICHIERS

Pour trouver un fichier, le système doit connaître :

- le nom du fichier,
- l'état du fichier,
- le nom du volume dans lequel est enregistré le fichier,
- et l'identification de l'appareil sur lequel est monté le volume contenant le fichier.

Ces informations doivent identifier le fichier de façon unique.

#### 3.1.1 Types de fichiers

Le système prend en charge les fichiers utilisateur et les fichiers bibliothèque, qu'ils soient permanents ou temporaires, ainsi que les fichiers SYSIN.

Les informations à indiquer pour l'identification d'un fichier dépendent de la nature permanente ou temporaire de ce fichier, et, s'il est permanent, de type catalogué ou non.

### 3.1.1.1 Fichiers catalogués, non catalogués et temporaires

Un fichier temporaire ou un fichier permanent non catalogué est identifié par la combinaison **unique** du nom et du statut du fichier, du nom du volume dans lequel il est enregistré et du nom de l'appareil sur lequel est monté ce volume. Si une seule de ces informations diffère entre deux identifications de fichier, le système considère qu'il s'agit de deux fichiers distincts. Aussi, il est indispensable de spécifier ces quatre informations pour permettre au système d'identifier correctement les fichiers.

**Exemples :**

```
ASSIGN MYFILE, PAYFILE.TRANS, DEVCLASS=MS/M500, MEDIA=B069;
ASSIGN MYFILE, PAYFILE.TRANS, DEVCLASS=MS/M500, MEDIA=B068;
ASSIGN MYFILE, PAYFILE.TRANS, DEVCLASS=MS/M350, MEDIA=B069;
```

Dans ces identifications, une information étant différente pour chaque cas, ces exemples font référence à trois fichiers distincts.

Pour les fichiers catalogués (qui sont permanents par définition), le nom de volume et l'identification d'appareil (qui constituent les **paramètres d'implantation**) sont enregistrés de façon permanente dans un fichier spécial appelé CATALOG ; aussi, l'utilisateur fournit uniquement le nom de fichier.

CATALOG permet également à l'utilisateur de mieux contrôler l'accès aux fichiers et leur mise à jour. Pour plus de détails sur les catalogues et leur utilisation, voir le *Manuel de gestion des catalogues*.

Pour un fichier qui sera affecté en temps que fichier temporaire ou catalogué, l'utilisateur peut employer la syntaxe volset à la place du nom de volume et de l'identification d'appareil (pour plus de détails, se reporter au chapitre 3 du *Guide utilisateur JCL*). L'utilisateur peut aussi faire appel à cette syntaxe pour un fichier d'entrée temporaire.

### 3.1.1.2 Fichiers bibliothèque

Une bibliothèque est un fichier comportant plusieurs sous-fichiers appelés **unités de bibliothèque**. Chaque unité de bibliothèque est identifiée par son nom. Les bibliothèques peuvent être permanentes (cataloguées ou non) ou temporaires.

### 3.1.1.3 Fichier SYS.IN

Le système utilise un fichier appelé SYS.IN pour l'enregistrement temporaire des rubriques de description d'entrées des travaux utilisateur. Chaque rubrique de description d'entrées, enregistrée comme un sous-fichier temporaire de SYS.IN (SYSIN), est identifiée par un nom de rubrique de description d'entrées.

### 3.1.2 Noms de fichiers

Le nom de fichier est le nom sous lequel un fichier est connu du système (tel qu'il est enregistré dans le label de fichier). Il s'agit du nom de fichier externe.

**Remarque :** Dans un programme COBOL, la clause SELECT...ASSIGN définit le nom de fichier interne. Ce nom, qui peut comporter jusqu'à huit caractères, est associé au nom de fichier externe par l'ordre ASSIGN (voir chapitre 4).

Les paragraphes qui suivent précisent les règles d'identification des différents types de fichiers :

- Fichiers permanents.
- Fichiers temporaires.
- Fichiers bibliothèque.
- Unités de bibliothèque.
- Rubriques de description d'entrées.
- Fichiers non standard.

#### 3.1.2.1 Noms de fichiers permanents

Les paragraphes ci-après concernent surtout les fichiers catalogués, bien que l'identification des fichiers non catalogués puisse obéir aux mêmes règles.

Pour plus de précisions sur l'identification des fichiers catalogués, voir le *Manuel de gestion des catalogues*.

#### Généralités

Un **nom de fichier permanent** est un nom formé de noms d'éléments et de séparateurs. Le nom de fichier permanent d'un fichier catalogué doit avoir la structure suivante :

nom-composé.nom-simple

Le nom-composé se présentant lui-même sous la forme :

nom-répertoire-maître [.nom-répertoire]...

Chaque nom d'élément du nom composé peut comporter jusqu'à 16 caractères (lettres, chiffres, trait d'union et trait de soulignement). Un nom d'élément peut être un nom simple, un nom de répertoire ou un nom de répertoire maître. Pour des raisons de compatibilité avec d'autres systèmes DPS, il est préférable qu'un nom d'élément ne dépasse pas 12 caractères et ne comporte pas de trait de soulignement.

La longueur totale d'un nom de fichier permanent ne doit pas dépasser 33 caractères, séparateurs (.) compris. Si un fichier a un numéro de génération de 11 caractères (longueur maximum), la longueur maximum effective du nom de fichier permanent est de 44 caractères.

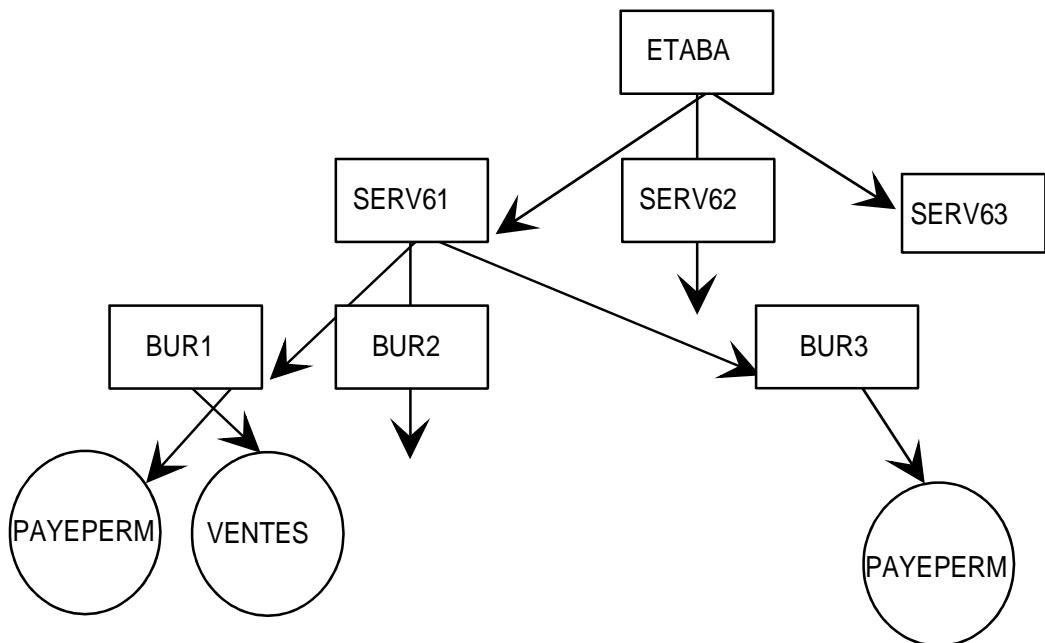
Un nom de fichier permanent reflète une hiérarchie de répertoires et de fichiers, organisée selon une structure arborescente, avec, au sommet, le répertoire maître, et aux niveaux inférieurs, des répertoires ou des fichiers.

**Exemple :**

Soit un fichier permanent PAYE, à traiter pour tenir compte des mouvements du BUREAU 1 du SERVICE 61 de l'ETABLISSEMENT A.

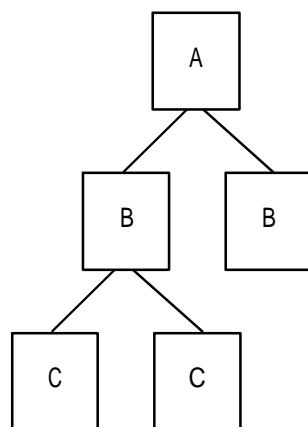
Son nom complet pourra être du type ETABA.SERV61.BUR1.PAYPERM et la structure arborescente pourra se présenter comme illustré par la figure 3-1.

PAYPERM est le nom simple de l'élément du niveau le plus bas (le fichier lui-même). BUR1, SERV61 et ETABA désignent les éléments des niveaux supérieurs.



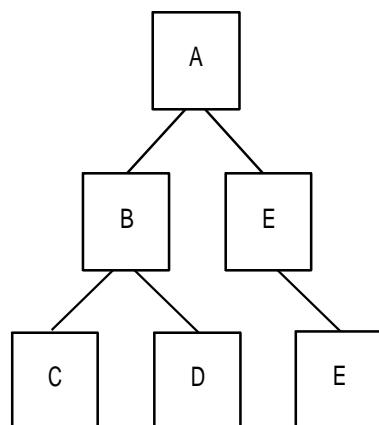
**Figure 3-1. Partie d'une structure arborescente**

Un nom d'élément doit être unique par rapport au nom d'élément de niveau immédiatement supérieur qui s'y rapporte. Il est donc interdit d'avoir deux noms identiques au même niveau inférieur à un nom d'élément donné. En revanche, si les deux noms identiques figurent sous deux noms d'éléments différents du même niveau ou se trouvent à des niveaux différents, ils sont acceptés. La figure 3-2 ci-dessous illustre cette règle.



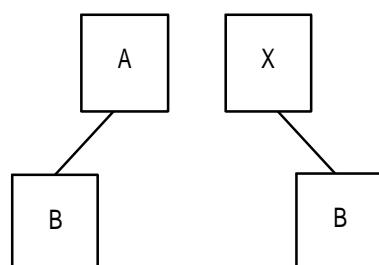
INCORRECT :

deux noms identiques  
sous un même nom  
de répertoire ou de  
répertoire maître



CORRECT :

deux noms identiques  
à des niveaux  
différents



CORRECT :

deux noms identiques  
sous des noms de  
répertoire différents

**Figure 3-2. Règles d'unicité des noms**

En résumé, les règles d'identification d'un fichier permanent catalogué utilisé sur DPS 7000 sont les suivantes :

- La longueur maximum d'un nom d'élément est de 12 caractères.
- Un nom d'élément doit commencer par une lettre (de A à Z), et ses autres caractères peuvent être des lettres, des chiffres (de 0 à 9) ou le trait d'union (-).
- Le point (.) sert de séparateur entre les différents noms d'éléments.
- Le nombre maximum de caractères autorisé pour un nom de fichier permanent est de 33 (points compris), plus 11 caractères pour l'éventuel numéro de génération.

### **Fichiers bande ANS**

Le nom d'un fichier bande ANSI permanent a une longueur maximum de 17 caractères (plus 11 caractères pour le numéro de génération éventuel si le fichier est catalogué). Les autres règles générales d'identification s'appliquent à ce type de fichiers suivant les règles énoncées plus haut.

### **Préfixage automatique**

En général, le nom de répertoire maître est le nom de projet que les utilisateurs indiquent pour les travaux devant y accéder.

Le mécanisme de préfixage automatique permet de ne pas mentionner le nom de projet dans le nom externe d'un fichier catalogué cité par un ordre JCL, le système ajoutant automatiquement le nom de projet pour le travail en cours (ou le nom indiqué par un ordre PREFIX antérieur) au début de tous les noms de fichiers externes commençant par un point.

Ainsi, si un travail s'exécute sous le nom de projet AREA3, l'ordre

ASSIGN INFILE, .INTL STOCK;

affecte le fichier catalogué AREA3.INTL STOCK au nom de fichier interne INFILE.

L'ordre PREFIX permet de définir un préfixe composé de plusieurs noms d'éléments.

### **Remarques sur les noms de fichiers non catalogués**

Un fichier non catalogué existe indépendamment de tout catalogue ou de toute structure hiérarchique. Il est identifié de façon unique par son nom et ses paramètres d'implantation (nom de volume et identification d'appareil). Il est donc sans importance que le nom d'un fichier non catalogué ne reflète pas la structure arborescente décrite pour les fichiers catalogués. Il est néanmoins préférable de désigner un fichier non catalogué comme s'il faisait partie d'une structure hiérarchique pour pouvoir, par exemple, l'incorporer ultérieurement au catalogue de site ou à un catalogue utilisateur.

***Exemples de noms de fichiers non catalogués :***

XTRAFILE1A

MY.FILE

HQ.ARCH.MY-FILE

Le deuxième et le troisième nom sont également valables pour des fichiers catalogués.

Les fichiers non catalogués ne peuvent avoir de numéro de génération. Autrement dit, la longueur d'un nom de fichier non catalogué ne peut pas dépasser 44 caractères (lettres, chiffres, trait d'union, point). Il est toutefois conseillé, en pratique, de ne pas dépasser 33 caractères, ce qui correspond à la longueur maximum d'un nom de fichier catalogué sans numéro de génération.

**Volumes bande multifichiers**

Un volume bande standard contenant plusieurs fichiers est dit multifichier. Le paramètre FSN de l'ordre ASSIGN peut être utilisé pour indiquer le rang (c'est-à-dire la position relative) du fichier dans le volume (par exemple, FSN=3 signifie qu'il s'agit du 3e fichier). Un volume bande multifichier ne peut pas contenir de fichier catalogué ; un fichier bande catalogué doit donc se trouver sur un volume monofichier. Les bandes multifichiers non standard ne sont acceptées que si elles n'ont pas de label (LABEL=NONE) ; en cas de label non standard (LABEL=NSTD), elles ne sont acceptées qu'en entrée.

**3.1.2.2 Noms de fichiers temporaires**

Un fichier temporaire est un fichier qui existe seulement le temps d'une activité ou d'un travail. Il est automatiquement supprimé par le système en fin d'activité ou de travail.

**Remarque :** Le nom d'un fichier temporaire doit être conforme aux règles d'écriture des noms simples (16 caractères maximum), sauf pour les fichiers bande ANSI (9 caractères maximum).

**3.1.2.3 Noms de fichiers bibliothèques et d'unités de bibliothèque**

Les règles de formation des noms des fichiers bibliothèques sont identiques à celles des autres fichiers, énoncées précédemment.

Un nom d'unité de bibliothèque peut comporter jusqu'à 31 caractères (lettres de A à Z, chiffres de 0 à 9, trait d'union et caractère de soulignement) et doit commencer par une lettre ou un chiffre.

**3.1.2.4 Noms de rubriques de description d'entrées**

La longueur maximum d'un nom de rubrique de description d'entrées est de 16 caractères. Ce nom doit commencer par une lettre et les caractères suivants peuvent être des lettres, des chiffres, le trait d'union et le caractère de soulignement.

## 3.1.2.5 Noms de fichiers non standard

Les noms de fichiers non conformes aux règles DPS 7000 sont considérés comme non standard. Les noms de fichiers non standard doivent être encadrés par des apostrophes (').

**Exemple :**

'A+B'

## 3.1.2.6 Rappel

Le tableau 3-1 récapitule les différents types de noms de fichier (et éléments de noms de fichier), avec leurs longueurs maximum autorisées, en nombre de caractères. La longueur maximum conseillée pour chaque nom est également indiquée, pour des raisons de compatibilité. Noter que la longueur maximum des noms de fichiers permanents inclut tous les séparateurs (points).

**Tableau 3-1. Longueur maximum des noms de fichier**

Type de fichier	Longueur maximum théorique du nom de fichier	Longueur maximum conseillée du nom de fichier
Nom de fichier permanent (fichier catalogué)	33 (+11 pour le numéro de génération)	33 (+11 pour le numéro de génération)
Nom simple	16	12
Nom de fichier permanent (fichier non catalogué)	44	33
Nom de fichier temporaire	16	16
Nom de fichier permanent (fichier bande ANS)	17 (+11 pour le numéro de génération)	17 (+11 pour le numéro de génération)
Nom de fichier temporaire (fichier bande ANS)	9	9
Nom d'unité de bibliothèque (sous-fichier de bibliothèque)	31	31
Nom de rubrique de description d'entrées	16	16

### 3.1.3 Statut des fichiers

Le **statut d'un fichier** indique si celui-ci est catalogué, non catalogué ou temporaire. L'utilisateur peut le spécifier dans l'ordre ASSIGN comme suit :

- FILESTAT=CAT pour un fichier catalogué,
- FILESTAT=UNCAT pour un fichier non catalogué,
- FILESTAT=TEMPRY pour un fichier temporaire.

Il est théoriquement inutile de spécifier le paramètre à mot-clé FILESTAT dans ASSIGN, car :

- FILESTAT=UNCAT est implicite lorsque les paramètres d'implantation (DEVCLASS - ou DVIDLIST - et MEDIA ou RESIDENT) sont spécifiés.
- FILESTAT=CAT est implicite lorsqu'aucun paramètre d'implantation n'apparaît dans l'ordre ASSIGN et lorsqu'il existe une rubrique correspondant au fichier dans un catalogue explicitement associé au traitement.
- Le paramètre autodéfini TEMPARY peut toujours apparaître à la place du paramètre à mot-clé FILESTAT=TEMPRY dans l'ordre ASSIGN.

En pratique, il est néanmoins recommandé de toujours spécifier FILESTAT dans l'ordre ASSIGN afin d'éviter toute ambiguïté lorsqu'un fichier est conservé d'une activité à une autre (les paramètres d'implantation n'étant pas obligatoires dans ce cas, même pour un fichier non catalogué ou temporaire).

### 3.1.4 Identification des volumes

Les paragraphes qui suivent concernent uniquement les fichiers permanents non catalogués et les fichiers temporaires. En effet, l'identification des volumes ne s'utilise pas dans les ordres JCL relatifs aux fichiers catalogués, sauf dans certaines situations impliquant la gestion d'un catalogue (voir le *Manuel utilisateur de gestion des catalogues*).

Un volume peut être "connecté" en permanence au système ou monté sur un appareil uniquement lorsqu'il est nécessaire ; dans le premier cas, il est dit **résidant** et dans le second cas **non résidant**.

#### 3.1.4.1 Volume résidant

Un volume résidant est toujours un disque et est identifié par le paramètre RESIDENT. Il n'est pas nécessaire de spécifier d'identification d'appareil pour un volume résidant.

**Exemple :**

```
ASSIGN PAY, PAY.MASTER, RESIDENT;
```

Le fichier permanent non catalogué PAY.MASTER est implanté sur un volume disque résidant. Noter que FILESTAT=UNCAT est implicite lorsque RESIDENT est spécifié.

## 3.1.4.2 Volume non résidant

Le nom d'un volume disque non résidant, cartouche ou bande (enregistré dans le label de volume) doit être spécifié au moyen du paramètre MEDIA.

***Exemple :***

MEDIA=B016

Le nom de volume peut comporter des lettres, des chiffres, le trait d'union et le caractère de soulignement. Sa longueur maximum est de 6 caractères. Le mot-clé WORK indique un volume de travail cartouche ou bande.

***Exemple :***

MEDIA=WORK

Les bandes ou cartouches de travail se préparent au moyen de l'utilitaire VOLPREP ou TAPEPREP (voir chapitre 5).

Dans certains cas, l'identification d'un volume peut comporter le paramètre LABEL, pour indiquer le type de label de volume.

***Exemples :***

LABEL=NATIVE

indique un volume avec organisation de labels GCOS 7 standard (valeur implicite).

LABEL=NONE

indique un volume sans label.

(voir l'ordre ASSIGN pour les autres valeurs possibles de LABEL).

Les paramètres MEDIA et LABEL ne peuvent être spécifiés qu'en présence d'un paramètre d'identification d'appareil (DEVCLASS ou DVCLIST - voir paragraphe suivant). Pour plus de précisions sur le paramètre LABEL, voir l'ordre ASSIGN au chapitre 4.

## 3.1.4.3 Noms de volumes non standard

Les noms de volumes non conformes aux règles DPS 7000 exposées au paragraphe 3.1.4 sont considérés comme non standard.

Les noms de volumes non standard doivent être protégés, c'est-à-dire encadrés par des apostrophes (').

***Exemple :***

'A+B' .

Pour faire référence à des noms de volumes non standard contenant des caractères non imprimables (des zéros binaires, par exemple), l'utilisateur doit leur affecter :

```
MEDIA=*,  
DVIDLIST=(nom-appareil)
```

**Exemple :**

L'ordre suivant permet de monter une bande non désignée par un nom standard sur le dérouleur MT02 :

```
ASSIGN MYIFN, * , MEDIA=*, DVIDLIST=(MT02) ;
```

Pour plus de précisions sur les paramètres associés à l'ordre ASSIGN, voir cet ordre au chapitre 4.

### 3.1.5 Identification des appareils

Les paragraphes qui suivent concernent uniquement les fichiers permanents non catalogués et les fichiers temporaires. En effet, l'identification des volumes ne s'utilise pas dans les ordres JCL relatifs aux fichiers catalogués, sauf dans certaines situations impliquant la gestion d'un catalogue (voir le *Manuel utilisateur de gestion des catalogues*).

Un appareil s'identifie :

- soit au moyen du paramètre DEVCLASS indiquant sa classe d'appareils et ses attributs,
- soit au moyen du paramètre DVIDLIST indiquant son nom d'appareil.

#### 3.1.5.1 Classe et attributs des appareils

La **classe d'appareils** indique un type de périphérique : par exemple, unité de disques, dérouleur de bandes classiques ou dérouleur de bandes en cartouche. Elle peut être complétée par des **attributs**, qui précisent une caractéristique particulière de ce type d'appareils, une option technique ou une configuration spécifiques, etc.

La classe d'appareils est spécifiée sous la forme :

```
xx[ / attribut]...
```

xx représente un code à deux caractères alphabétiques, dont la signification est la suivante :

CT	Dérouleur de bandes en cartouche
MS	Unité de disques (mémoire de masse)
MT	Dérouleur de bandes classique
PR	Imprimante
TN	Terminal

Lorsque la classe d'appareils est mentionnée seule (XX), elle désigne n'importe quel appareil de ce type.

### ATTENTION

La classe d'appareils MS doit être accompagnée de ses attributs correspondants ; sinon, le système peut provoquer un arrêt prématuré de l'activité.

Pour les autres classes, les attributs supplémentaires peuvent être omis, à condition qu'ils ne présentent pas d'ambiguïté et qu'aucune caractéristique spécifique ne soit demandée.

A propos des dérouleurs de bandes en cartouche :

Pour les cartouches au format 8 mm, l'attribut M6 est facultatif si tous les dérouleurs de bandes utilisent des cartouches de ce format.

Pour les cartouches au format 1/2 pouce ne faisant pas partie d'une bibliothèque, les attributs M5 ou 36T sont facultatifs si tous les dérouleurs de bande utilisent des cartouches de ce format avec le //dame recording format// (18 ou 36 pistes) et s'il n'existe aucune bibliothèque.

Pour les appareils faisant partie d'une bibliothèque, les attributs M5 ou 36T sont facultatifs si tous les appareils de la bibliothèque utilisent le même format d'enregistrement (18 ou 36 pistes).

Il est néanmoins recommandé d'utiliser la totalité de la classe d'appareils ce qui permet d'éviter d'avoir à modifier les travaux ou les catalogues lors de l'introduction de nouveaux appareils sur le site ou bien lorsque les travaux sont lancés sur un autre site.

Une liste des classes d'appareils et de leurs attributs possibles figure au paragraphe 3.1.5.2.

#### **Exemples :**

```
ASSIGN IFN3, PAYDATA, TEMPRY, DEVCLASS=MS/D500, MEDIA=1234;
```

Le nom de fichier interne IFN3 est associé au fichier temporaire PAYDATA, enregistré sur un volume disque de 500 Mo dont le nom est 1234.

```
ASSIGN INC, MY.TAC, DEVCLASS=MT/T9/D1600, MEDIA=B016;
```

Cet ordre demande l'affection du fichier non catalogué MY.TAC, enregistré sur un volume bande 9 pistes de densité 1.600 bpi, le nom du volume étant B016. Noter que FILESTAT=UNCAT est implicite lorsque DEVCLASS et MEDIA sont spécifiés.

### 3.1.5.2 Liste des classes et attributs d'appareils existants

## Dérouleurs de bandes en cartouche (n'appartenant pas à une bibliothèque)

Noter qu'aucun attribut ne permet de distinguer un dérouleur utilisant un chargeur de cartouches SCL.

### Cartouches 8 mm :

CT [ /M6 ] [ / (S35 | S75) ]

où

S35 indique une densité d'enregistrement de 35 Mo/pouce ;  
S75 indique une densité d'enregistrement de 75 Mo/pouce.

### Cartouches 1/2 pouce :

CT [ / {M5 | 36T} ] [ /C ]

où

M5 spécifie que le dérouleur de bande utilise le format d'enregistrement à 18 pistes (seules les bandes de longueur standard sont acceptées). Noter que l'utilisation de cartouches formatées à 36 pistes sur un tel dispositif implique l'utilisation de l'attribut M5. Le seul usage possible est alors la préparation des volumes d'une bande de longueur standard (la cartouche sera toujours formatée à 18 pistes).

36T spécifie que le dérouleur de bande utilise le format d'enregistrement à 36 pistes. Aucun attribut ne permet de distinguer les bandes de longueur standard des bandes de format E (ces deux types de bande sont acceptés). Noter que l'utilisation d'une cartouche formatée à 18 pistes sur un tel dispositif implique l'utilisation de l'attribut 36T. Les différentes utilisations sont la lecture, la préparation des volumes (la cartouche sera toujours une cartouche formatée à 36 pistes), ou encore la réécriture (toujours au format d'enregistrement à 36 pistes).

5

spécifie le compactage. Noter que :

- les différents dérouleurs de bande qui utilisent le format d'enregistrement à 36 pistes disposent tous d'une capacité de compactage.
  - pour lire une cartouche écrite en mode compactage, l'attribut C doit être spécifié (sauf si, sur le site, tous les dispositifs du même type utilisent ce mode).
  - pour écrire sur une cartouche en mode compactage, l'attribut C doit être spécifié.
  - pour lire une cartouche écrite en mode non-compacté, l'attribut C peut être, ou non, spécifié (si l'attribut C est spécifié, cette opération doit être réalisée à partir d'un dérouleur autorisant le compactage ; sinon la capacité de compactage n'est pas obligatoire).

- pour écrire sur une cartouche en mode non-compacté, on ne doit pas spécifier l'attribut C (cependant, cette opération peut être réalisée à partir d'un dérouleur de bande disposant d'une capacité de compactage).

**Exemples :**

CT	bande en cartouche.
CT/M6	bande en cartouche 8 mm.
CT/M6/S35	bande en cartouche 8 mm d'une densité de 35 Mo/pouce <sup>2</sup> .
CT/M5	bande en cartouche 1/2 pouce (utilisant un format d'enregistrement à 18 pistes).
CT/36T/C	bande en cartouche 1/2 pouce (utilisant un format d'enregistrement à 36 pistes) avec compactage.

**Bibliothèque de cartouches (format 1/2 pouce uniquement)**

CT/LIB [/ {M5 | 36T}] [/C]

où

M5, 36T et C ont la même signification que pour les cartouches au format 1/2 pouce n'appartenant pas à une bibliothèque de cartouches.

**Exemples :**

CT/LIB	bande en cartouche appartenant à une bibliothèque.
CT/LIB/C	bande en cartouche avec compactage, appartenant à une bibliothèque.
CT/LIB/M5/C	bande en cartouche avec compactage, appartenant à une bibliothèque (utilisant un format d'enregistrement à 18 pistes)
CT/LIB/36T	bande en cartouche appartenant à une bibliothèque (utilisant un format d'enregistrement à 36 pistes)

**Unités de disque**

MS/{FSA|B10|D500}

**Exemples :**

MS/FSA	Unité de disques à secteurs fixes
MS/B10	Unité de disques de 1 Go.
MS/D500	Unité de disques de 500 Mo.

### Dérouleurs de bandes classiques

MT [ /T9 ] [ /Dnnnn ] [ /S ]

T9	nombre de pistes (9).
Dnnnn	densité (D1600 ou D6250).
S	mode continu. Ce mode accélère la procédure si le traitement est effectué sans arrêter la bande. Ce mode est un mode de traitement, et non une caractéristique du volume de la bande concernée. S n'a de signification que lorsque la commande concernée est FILREST, FILSAVE, VOLREST ou VOLSAVE. Dans tous les autres cas, on utilise le mode //start/stop//.

#### *Exemples :*

MT/T9	Unité de bandes 9 pistes à densité non spécifiée.
MT/D1600	Unité de bandes 9 pistes de 1 600 bits/pouce.
MT/T9/D6250	Unité de bandes 9 pistes de 6 250 bits/pouce.
MT/T9/D6250/S	Unité de bandes 9 pistes de 6 250 bits/pouce fonctionnant en mode continu.
MT/S	Unité de bandes 9 pistes à densité non-spécifiée fonctionnant en mode continu.

### Imprimantes

PR [ /vitesse ] [ /Hnnn ] [ /modèle ]

vitesse attribut de vitesse (FI, A ou SI). Voir remarque 2 ci-après.

Hnnn imprimante capable de gérer au moins nnn positions d'impression (par exemple, PR/H125 correspond à une imprimante à 132, 136 ou 160 marteaux, mais non à 120 marteaux).

modèle modèle de l'imprimante (PR54, PR90).

#### *Exemple :*

PR

N'importe quelle imprimante.

PR/H123

Imprimante à 123 marteaux minimum.

- Remarques :**
1. Si l'attribut Hnnn est omis, l'imprimante retenue peut être n'importe laquelle (selon les autres attributs spécifiés - voir ci-dessous).
  2. Un attribut de vitesse d'impression peut être spécifié :  
SI correspond à l'imprimante la plus lente compte tenu du nombre de marteaux éventuellement spécifié.  
FI correspond à l'imprimante la plus rapide compte tenu du nombre de marteaux éventuellement spécifié.  
A correspond à une vitesse quelconque.
  3. Il est possible de spécifier un type d'imprimante particulier (par exemple, PR54).

## Terminaux

TN (utilisé dans ::TN pour file terminal)

### 3.1.5.3 Nom d'appareil

Le nom d'appareil est le nom désignant un périphérique précis. Les noms d'appareils sont affectés à la génération du système.

Le format d'un nom d'appareil est le suivant :

XXYY

XX correspondant au type de périphérique (MT, MS, etc.) et YY étant un identificateur formé de deux caractères alphanumériques.

Les noms d'appareils ne doivent pas être indiqués pour des volumes ou des disques miroir résidants.

Par ailleurs, la reconfiguration d'un appareil n'est pas possible si un nom d'appareil est indiqué.

### *Exemple :*

CT01

- Remarque :** Le paramètre DVIDLIST accompagné du nom d'appareil sert à demander un appareil précis. Si cet appareil est en cours d'utilisation par un autre travail (c'est-à-dire non disponible), la demande est placée en file d'attente jusqu'à libération de la ressource. L'utilisation du paramètre DEVCLASS (classe et attributs d'appareils) est donc préférable.

## 3.2 GROUPES DE PARAMETRES STANDARD

Certains groupes de paramètres figurent à l'identique dans plusieurs ordres JCL étendus. Afin de ne pas avoir à répéter la totalité de chaque groupe dans le format des ordres concernés, chacun de ces groupes est développé dans le présent chapitre et désigné par un nom standard, qui est ensuite mentionné seul dans les ordres. Ce nom est également utilisé dans les autres manuels décrivant des ordres JCL.

La notation définie dans le tableau 3-2 décrit la ou les valeurs pouvant être prise(s) par chaque paramètre. Cette notation est utilisée dans tout le manuel.

**Tableau 3-2. Valeur des paramètres**

Notation	Description
alphanum	chaîne de caractères alphanumériques (lettres de A à Z, chiffres de 0 à 9, trait d'union, caractère de soulignement) ; le premier caractère ne doit pas être un trait d'union ou un caractère de soulignement.
alphanumN	chaîne alphanumérique d'une longueur maximum de N caractères.
alphanum_N	chaîne alphanumérique d'une longueur de N caractères.
bit	chaîne de chiffres binaires (0, 1).
bitN	chaîne de chiffres binaires d'une longueur maximum de N chiffres.
bit_N	chaîne de chiffres binaires d'une longueur de N chiffres.
chiffres	chaîne de chiffres décimaux (0 à 9).
chiffresN	chaîne de chiffres décimaux d'une longueur maximum de N chiffres.
chiffres_N	chaîne de chiffres décimaux d'une longueur de N chiffres.
hexa	chaîne de caractères hexadécimaux (0 à 9, A à F).
hexaN	chaîne de caractères hexadécimaux d'une longueur maximum de N caractères.
hexa_N	chaîne de caractères hexadécimaux d'une longueur de N caractères.
identificateur	chaîne de caractères alphanumériques commençant par une lettre (A à Z).
identificateurN	identificateur d'une longueur maximum de N caractères.
identificateur_N	identificateur d'une longueur de N caractères.
octal	chaîne de caractères octaux (0 à 7).
octalN	chaîne de caractères octaux d'une longueur maximum de N octaux.
octal_N	chaîne de caractères octaux d'une longueur de N octaux.
chaîne	chaîne de caractères quelconques.
chaîneN	chaîne de caractères quelconques d'une longueur maximum de N caractères.
chaîne_N	chaîne de caractères quelconques d'une longueur de N caractères.

N représente un entier positif : 1, 2, 3, 4, etc.

Le tableau 3-3 ci-dessous présente la syntaxe des noms apparaissant dans les ordres JCL, en utilisant la notation détaillée dans le tableau 3-2.

**Tableau 3-3. Syntaxe des noms figurant dans les ordres JCL**

Notation	Description
nom-compte	alphanum12
nom-catalogue	nom-simple [.nom-simple]. CATALOG
nom-fichier-externe	nom-qualifié (longueur < 44 car.) chaîne protégée
nom-rubrique-description-d'entrées	alphanum16
nom-fichier-interne	alphanum8
nom-travail	alphanum8
nom-label	alphanum8
nom-module-chargeable	alphanum31
nom-répertoire-maître	nom-simple
nom-volume	alphanum6 (longueur < 6 car.) chaîne protégée
nom-unité-bibliothèque	alphanum31
nom-projet	alphanum12
nom-qualifié	nom-simple [.nom-simple]...
nom-simple	alphanum16
nom-sous-convention-astérisque	alphanum31 dont les parties non spécifiées sont remplacées par un astérisque
nom-fichier-temporaire	nom-simple
nom-utilisateur	alphanum12

Les noms des groupes de paramètres standard et leurs formats figurent dans les paragraphes qui suivent.

### 3.2.1 Description-fichier-d'entrée-séquentiel

#### Format

```

{nom-rubrique-description-d'entrées}      [{FILESTAT = {CAT      }}]
{nom-fichier-externe }                   [{      {UNCAT    }}]
{      *          }                   [{      {TEMPRY  }}]
{      {TEMPRY  }}]

[ {CATALOG = chiffre 1
[ {RESIDENT
[ {DEVCLASS = classe-d'appareils MEDIA = {WORK
[ {      *          }
[ {      { (nom-volume[nom-volume]...)
[ {VOLSET = {nom-volset|DFLT}

[SUBFILE = nom-unité-bibliothèque]

[SHARE = {NORMAL  } ]      [ACCESS = {WRITE   }]
[ {ONEWRITE} ]           [{SPWRITE  }]
[ {FREE    } ]           [{READ     }]
[ {MONITOR } ]           [{SPREAD   }]
[ {DIR     } ]           [{RECOVERY}]
[ {          } ]           [{ALLREAD }]

[FSN     = {chiffres3} ]
[ {ANY    } ]
[ {NEXT   } ]

[NBEFN   = {chiffres3} ]      [FIRSTVOL = {chiffres3} ]
[ {ALL    } ]           [{EOF      }]

[LASTVOL = {chiffres3} ]      [END      = {DEASSIGN}]
[ {EOF    } ]           [{PASS     }]
[ {          } ]           [{LEAVE    }]
[ {          } ]           [{UNLOAD   }]

[ABEND   = {DEASSIGN} ]
[ {PASS    } ]           [MOUNT     = chiffres2]
[ {LEAVE    } ]
[ {UNLOAD   } ]

[POOL    [ {FIRST} ] ]      [LABEL     = {NATIVE  }]
[ {NEXT   } ]           [{NONE     }]
[ {          } ]           [{NSTD     }]

```

#### Commentaires et restrictions

SYSOUT et DUMMY ne peuvent être spécifiés comme noms de fichiers externes. Le nom de fichier externe peut être TEMP, TEMP1 ou TEMP2, mais dans ce cas, seul le paramètre SUBFILE doit être spécifié.

Le fichier concerné est obligatoirement séquentiel.

Si SUBFILE est spécifié, SHARE a implicitement la valeur DIR.

La valeur implicite de ACCESS est READ (noter que dans l'ordre ASSIGN, la valeur implicite de ACCESS est WRITE).

Les autres paramètres et leurs valeurs sont les mêmes que dans l'ordre ASSIGN.

### 3.2.2 Description-fichier-d'entrée

Ce groupe de paramètres est identique à la description du fichier d'entrée séquentiel, à la seule différence que le fichier concerné n'est pas nécessairement séquentiel.

### 3.2.3 Description-fichier-de-sortie-séquentiel

## Format

```

{nom-fichier-externe}      [{FILESTAT  = { CAT      }}]
{DUMMY                      [{UNCAT     }]}
{                           [{TEMPRY    }]}
{                           [{TEMPRY    }]}]

[ { CATALOG   = chiffrel
[ { RESIDENT
[ { DEVCLASS  = classe-d'appareils MEDIA  = {WORK
[ {                           {*
[ {                           { (nom-volume [nom-volume]...)
[ { VOLSET    = {nom-volset | DFLT }]

[ { FSN        = {chiffres3}
[ {                           {ANY      }
[ {                           {NEXT     }]}]

[ { SUBFILE   = nom-unité-bibliothèque] [ {NEXT     }]}]

[ { SHARE     = {NORMAL   } ] [ {ACCESS   = {WRITE    } }
[ {                           {ONEWRITE } ] [ {          {SPWRITE   } }
[ {                           {FREE     } } [ {          {READ     } }
[ {                           {DIR      } } [ {          {SPREAD    } }
[ {                           {MONITOR  } } [ {          {RECOVERY  } }
[ {                           ] [ {          {ALLREAD  } }]

[ { END       = {DEASSIGN} ] [ {ABEND    = {DEASSIGN} }
[ {                           {PASS     } } [ {          {PASS     } }
[ {                           {LEAVE    } } [ {          {LEAVE    } }
[ {                           {UNLOAD   } } [ {          {UNLOAD   } }]

[ { LABEL     = {NATIVE   } ] [ {MOUNT    = chiffres2}
[ {                           {NONE    } } [ {          {FIRSTVOL = {chiffres3}
[ {                           {NSTD    } } [ {          {EOF      } } ]
[ {                           ] [ {          {LASTVOL  = {chiffres3}
[ {                           ] [ {          {EOF      } } ] }

[ { EXPDATE   = {jjj      } ] [ {DENSITY  = {D1600} }
[ {                           {aa/jjj   } } [ {          {D6250} }
[ {                           {aa/mmm/jjj} } [ {          {S35     } }
[ {                           ] [ {          {S75     } } ] }

[ { SIZE      = chiffres3] [ {POOL    [ {FIRST} ] ]
[ {                           [ {NEXT   } ] ]
[ {                           ] ] }

[ {CATNOW]  [ {CATALOG=chiffrel} ]

```

## Commentaires et restrictions

La signification de chaque paramètre du groupe, à l'exception de SIZE, est donnée avec l'ordre ASSIGN.

La valeur implicite de ACCESS est WRITE

Le paramètre SIZE indique le nombre de cylindres de disque à réservé au fichier ; il ne peut être spécifié que pour les fichiers temporaires.

### 3.2.4 Description-fichier-de-sortie

Ce groupe de paramètres est identique à la description de fichier de sortie séquentiel, à la seule différence que le fichier concerné n'est pas nécessairement séquentiel.

### 3.2.5 Description-fichier-d'impression

Les paramètres de ce groupe sont identiques à ceux de la description de fichier de sortie séquentiel, aux deux différences près que SYS.OUT peut être utilisé comme nom de fichier externe, mais non TEMP, TEMP1 et TEMP2.

### 3.2.6 Description-fichier-de-travail

#### Format

```

[ {FILESTAT = {CAT      } } ]
[ {      {UNCAT    } } ]
nom-fichier-externe [ {      {TEMPRY   } } ]
[ {TEMPRY      } ] }

[ {CATALOG  = chiffrel
[ {RESIDENT
[ {DEVCLASS = classe-d'appareils MEDIA = {WORK
[ {                           (nom-volume [nom-volume]...)
[ {VOLSET  = {volset-name | DFLT} } ] }

[ SHARE      = {NORMAL    } ] [ ACCESS   = {WRITE     } ]
[ {ONEWRITE  } ] [ {SPWRITE   } ]
[ {FREE      } ] [ {READ      } ]
[ {DIR       } ] { {SPREAD    } ]
[ {MONITOR   } ] [ {ALLREAD   } ]

[ LABEL      = NATIVE ]      [ SIZE     = chiffres3]
[ END        = {DEASSIGN} ] [ ABEND    = {DEASSIGN} ]
[ {PASS      } ] [ {PASS      } ] }

[ EXPDATE   = { jjj      } ]
[ {aa/jjj    } ]
[ {aa/mm/jj} ]

```

#### Commentaires et restrictions

Le paramètre SIZE ne peut être spécifié que pour un fichier temporaire ; il indique alors le nombre de cylindres disque à réserver au fichier. S'il est omis, le fichier de travail doit avoir fait l'objet d'une réservation statique de place (par exemple, au moyen de l'utilitaire PREALLOC).

Les autres paramètres sont décrits avec l'ordre ASSIGN.

### 3.2.7 Description-bibliothèque-d'entrée

## Format

```

{
{nom-fichier-externe} } [ {FILESTAT = {CAT      }} ]
{TEMP} } [ {          {UNCAT    }} ]
{TEMP1} } [ {          {TEMPRY   }} ]
{TEMP2} } [ {          {TEMPRY   }} ]
[ {CATALOG = chiffrel
[ {RESIDENT
[ {DEVCLASS= classe-d'appareils MEDIA=(nom-volume [nom-volume]...)}
[ {VOLSET = {nom-volset | DFLT}
[ {SHARE      = {NORMAL    }} ] [ {ACCESS     = {WRITE     }} ]
[ {          {ONEWRITE  }} ] [ {          {SPWRITE   }} ]
[ {          {FREE      }} ] [ {          {READ      }} ]
[ {          {DIR       }} ] [ {          {SPREAD    }} ]
[ {          } ] [ {          {ALLREAD  }} ]
[ {END        = {DEASSIGN}} ] [ {ABEND      = {DEASSIGN}} ]
[ {          {PASS      }} ] [ {          {PASS      }} ]
[ {LABEL      = NATIVE} ]

```

## Commentaires et restrictions

SYSOUT, DUMMY et les rubriques de description d'entrées ne doivent pas être spécifiés comme noms de fichiers externes. Si le nom de fichier externe est TEMP, TEMP1 ou TEMP2, le système force le paramètre FILESTAT à la valeur TEMPRY, et les paramètres END et ABEND prennent tous deux la valeur implicite PASS (au lieu de DEASSIGN).

Les autres paramètres et leurs valeurs sont les mêmes que dans l'ordre ASSIGN.

### 3.2.8 Description-bibliothèque-de-sortie

#### Format

```

{nom-fichier-externe}      [{FILESTAT = {CAT      }}]
{TEMP}                      [{      {UNCAT      }}]
{TEMP1}                     [{      {TEMPRY      }}]
{TEMP2}                     [{      {TEMPRY      }}]

[ {CATALOG = chiffrel      }]
[ {RESIDENT      }]
[ {DEVCLASS=classe-d'appareils MEDIA=(nom-volume[nom-volume]...)      }]
[ {VOLSET = {nom-volset | DFLT }}      ]

[SHARE      = {NORMAL      }]  [ACCESS      = {WRITE      }]
[      {ONEWRITE      }]  [      {SPWRITE      }]
[      {FREE      }]  [      {READ      }]
[      {DIR      }]  [      {SPREAD      }]
[      ]  [      {RECOVERY      }]

[END      = {DEASSIGN      }]  [ABEND      = {DEASSIGN      }]
[      {PASS      }]  [      {PASS      }]

[LABEL      = NATIVE]  [SIZE      = chiffres3]

[EXPDATE      = {jjj      }]
[      {aa/jjj      }]
[      {aa/mm/jj}      ]

```

#### Commentaires et restrictions

SYSPUT et DUMMY ne doivent pas être spécifiés comme noms de fichiers externes. Si le nom de fichier externe est TEMP, TEMP1 ou TEMP2, le paramètre FILESTAT est forcé à la valeur TEMPRY et les paramètres END et ABEND prennent tous deux la valeur implicite PASS (au lieu de DEASSIGN).

La valeur implicite de ACCESS est WRITE.

Le paramètre SIZE indique le nombre de cylindres disque à réserver au fichier (valable uniquement pour les fichiers temporaires).

Les autres paramètres et leurs valeurs sont les mêmes que dans l'ordre ASSIGN.

### 3.2.9 Description-bibliothèque-d'impression

Ce groupe de paramètres est identique à la description de bibliothèque de sortie.

TEMP, TEMP1 et TEMP2 peuvent être spécifiés comme noms de fichiers externes en mode interactif. Si aucun nom n'est spécifié, le nom de fichier externe implicite est TEMP.

### **3.2.10 Paramètres-define**

Il s'agit des paramètres de l'ordre JCL de base DEFINE. Voir ce dernier pour le format et la description de ces paramètres.

### **3.2.11 Paramètres-sysout**

Il s'agit des paramètres de l'ordre JCL de base SYSOUT. Voir ce dernier pour le format et la description de ces paramètres.

### **3.2.12 Paramètres-size**

Il s'agit des paramètres de l'ordre JCL de base SIZE. Voir ce dernier pour le format et la description de ces paramètres.

### **3.2.13 Paramètres-step**

Il s'agit des paramètres de l'ordre JCL de base STEP. Voir ce dernier pour le format et la description de ces paramètres.

OPTIONS, DEBUG et DUMP=ALL ne sont pas utilisables.

## 4. Ordres JCL de base

### 4.1 PRESENTATION DES ORDRES

Les informations relatives aux ordres JCL, présentés par ordre alphabétique, sont organisées comme suit :

1. Fonction
2. Rubrique
3. Format
4. Description
5. Paramètres
6. Exemples

#### 4.1.1 Fonction

Décrit brièvement le rôle de l'ordre.

#### 4.1.2 Rubrique

Indique à quel type de rubrique appartient l'ordre : travail, activité ou entrées.

### 4.1.3 Format

Fournit tous les paramètres de l'ordre concerné. Le modèle de format est présenté dans une police différente du reste du texte, et obéit aux conventions d'écriture suivantes :

ELEMENT	Les majuscules indiquent un mot-clé, à introduire tel quel.
élément	Les minuscules indiquent une valeur à fournir par l'utilisateur.
[ élément ]	Les crochets indiquent un élément facultatif.
{ élément1 } { élément2 } { élément3 }	Une colonne d'éléments entre accolades indique que l'un d'eux doit être sélectionné, si le paramètre correspondant est retenu. Si ce paramètre n'est pas retenu, c'est la valeur soulignée que le système adopte implicitement.
( )	Les parenthèses sont obligatoires si elles renferment plus d'un élément positionnel.
...	Les points de suspension indiquent que l'élément peut être répété une ou plusieurs fois.

### 4.1.4 Description

Fournit une description plus détaillée de l'ordre et comporte éventuellement un sous-titre, "Commentaires et restrictions".

### 4.1.5 Paramètres

Si l'ordre comporte des paramètres, décrit chaque paramètre (en premier lieu les paramètres obligatoires, et en second lieu les paramètres facultatifs).

### 4.1.6 Exemples

Des exemples illustrent la syntaxe de l'ordre.

## 4.2 ALLOCATE

### 4.2.1 Fonction

Demande l'affectation de place à un fichier disque ou disquette et/ou l'affectation d'une certaine place supplémentaire chaque fois que la place existante est occupée en entier.

### 4.2.2 Rubrique

Description d'activité.

### 4.2.3 Format

```
ALLOCATE nom-fichier-interne
  {
    {SIZE      = chiffres10
    {
      {INCRSIZE = chiffres5
      {
        {SIZE      = chiffres10  INCRSIZE = chiffres5
        {
          [UNIT      = {TRACK      } ]
          [
            {TRK       } ]
          [
            {CYLINDER  } ]
          [
            {CYL       } ]
          [
            {RECORD    } ]
          [
            {BLOCK    } ]
          [
            {100 KB    } ]
          [ CHECK ] ;
          [ KEEP ] ;
        }
      }
    }
  }
```

#### 4.2.4 Description de l'ordre

L'ordre ALLOCATE réserve de la place sur disque pour un fichier temporaire ou permanent non catalogué affecté à une activité. Pour les fichiers UFAS séquentiels et indexés et les fichiers bibliothèques, l'ordre ALLOCATE peut également demander l'**extension de fichier**, c'est-à-dire l'agrandissement de l'espace disque réservé au fichier. Tout ordre ALLOCATE doit être associé (au moyen du nom de fichier interne) à un ordre ASSIGN dans la même rubrique de description d'activité.

S'il existe un ordre ASSIGN pour un fichier temporaire à créer mais pas d'ordre ALLOCATE correspondant, un cylindre d'espace disque et, dans les cas appropriés, un cylindre d'extension, sont implicitement réservés au fichier.

Si, dans la même activité, plusieurs ordres ASSIGN portent sur le même nom de fichier permanent, l'ordre ALLOCATE n'est valable que s'il est associé au nom de fichier interne du premier ordre ASSIGN. Pour plus de détails sur l'affectation d'un même fichier plusieurs fois dans une activité, voir l'ordre ASSIGN (paragraphe 4.3).

Un ordre ALLOCATE est nécessaire lorsque le paramètre CATNOW est spécifié dans l'ordre ASSIGN, si le fichier catalogué à créer se trouve sur disque (ne s'applique pas aux fichiers bande).

La réservation de place ne devient effective qu'au moment de l'ouverture en sortie du fichier cité dans ALLOCATE (nom de fichier interne) ; si aucune tentative d'ouverture du fichier n'a lieu pendant l'exécution de l'activité, la réservation de place n'est pas effectuée. Toute tentative d'accès au fichier avant que la réservation de place soit effective génère un code d'erreur.

La réservation de place pour des fichiers permanents doit normalement être demandée au moyen de l'ordre JCL étendu PREALLOC (voir chapitre 5). L'ordre ALLOCATE ne peut pas être utilisé pour des fichiers IDS. La réservation de place pour des fichiers bibliothèques permanents doit normalement être demandée par un ordre JCL étendu LIBALLOC.

Lorsque le paramètre INCRSIZE est spécifié, l'éventuelle extension de fichier a lieu pendant les opérations d'écriture (extension dynamique), et non à l'ouverture du fichier. Ce paramètre ne s'applique pas aux fichiers disquette.

La réservation de place pour un fichier déjà existant provoque normalement l'émission du code retour ALREADY à l'ouverture du fichier, et la réservation n'est pas prise en compte. Si le paramètre CHECK a été spécifié dans l'ordre ALLOCATE, l'activité est arrêtée prématurément.

## 4.2.5 Paramètres

### 4.2.5.1 Paramètre obligatoire

#### **nom-fichier-interne**

Nom du fichier pour lequel est effectuée la réservation de place. C'est le nom qui doit figurer dans l'ordre ASSIGN correspondant. Il doit comporter huit caractères maximum (lettres, chiffres, trait d'union et caractère de soulignement), et commencer par un chiffre ou une lettre.

### 4.2.5.2 Paramètres facultatifs

#### **SIZE**

Indique la longueur du fichier en nombre d'unités d'affectation (voir le paramètre UNIT). Sa valeur implicite est 0. Le paramètre SIZE est obligatoire en l'absence de INCRSIZE.

#### **INCRSIZE**

Indique le nombre d'unités d'affectation (voir le paramètre UNIT) ajoutées à un fichier de sortie lors de chaque extension automatique dans l'activité en cours. Le paramètre INCRSIZE doit être spécifié en l'absence de SIZE. Sa valeur implicite est 0, c'est-à-dire qu'il n'y a pas d'extension automatique si INCRSIZE n'est pas mentionné. INCRSIZE ne s'applique pas aux fichiers disquette.

#### **UNIT**

Indique l'unité d'affectation de place ou d'extension : piste (TRACK, valeur par défaut), cylindre (CYLINDER), article (RECORD), bloc (BLOCK) ou quantum (100KB).  
Les abréviations suivantes peuvent être utilisées :  
TRK, pour piste.  
CYL, pour cylindre.

#### **CHECK**

S'il existe déjà un fichier portant le même nom de fichier externe que celui indiqué dans l'ordre ASSIGN associé, l'utilisation du paramètre CHECK provoque l'arrêt prématuré de l'activité en cours au moment de l'ouverture du fichier. Si CHECK n'est pas spécifié, le code retour ALREADY est émis.

#### **KEEP**

Si KEEP est spécifié, les paramètres d'affectation sont conservés puis appliqués à tous les fichiers qui se sont vus réservé de la place avec le même nom de fichier interne.

#### **Exemples :**

```
ALLOCATE BAC, SIZE=20, UNIT=CYL;
ALLOCATE FULL, INCRSIZE=2;
```

## 4.3 ASSIGN

### 4.3.1 Fonction

Affecte un fichier, c'est-à-dire le fait connaître à une activité, le lui rend accessible et alloue à cette activité les ressources associées au fichier.

### 4.3.2 Rubrique

Description d'activité.

### 4.3.3 Format

```

ASSIGN nom-fichier-interne

{nom-fichier-externe
{nom-fichier-temporaire
{*nom-rubrique-de-description-d'entrées
{DUMMY
{*}

[ {FILESTAT  =  {CAT          }] }
[ {                  {UNCAT      }] }
[ {                  {TEMPRY    }] }
[ {                  }] }
[ {TEMPRY      }] }

[ CATALOG    =  chiffre1]

[ CATNOW  ]

[ EXPDATE   =  {jjj          }] ]
[ {                  {aa/jjj     }] }
[ {                  {aa/mm/jj}}]

[ {VOLWR   }] ]
[ {NVOLWR  }] }

[ {OPTIONAL}]

[ DEFER  ]

[ {RESIDENT          }] ]
[ {description-classe-d'appareils  }] ]
[ {liste-identifications-d'appareils  }] ]
[ {VOLSET = {nom-volset|DFLT}}]

[ SUBFILE  = nom-unité-bibliothèque      ]
```

## Ordres JCL de base

```

[ SHARE      = {NORMAL      } ]
[             {ONEWRITE    } ]
[             {FREE        } ]
[             {MONITOR     } ]
[             {DIR         } ]

[ ACCESS     = {WRITE       } ]
[             {READ        } ]
[             {SPWRITE     } ]
[             {SPREAD      } ]
[             {RECOVERY    } ]
[             {ALLREAD    } ]

[ END        = {PASS        } ]
[             {DEASSIGN    } ]
[             {LEAVE        } ]
[             {UNLOAD      } ]

[ ABEND      = {PASS        } ]
[             {DEASSIGN    } ]
[             {LEAVE        } ]
[             {UNLOAD      } ]

[ ENDVOL     = {UNLOAD     } ]
[             {REWIND      } ]
[             {LEAVE        } ]

[ MOUNT      = chiffres2  ]

[ POOL       = {FIRST} ]
[             {NEXT } ]

[ LABEL      = {NATIVE} ]
[             {NONE  } ]
[             {NSTD  } ]

[ FIRSTVOL  = {chiffres3} ]
[             {EOF   } ]

[ LASTVOL   = {chiffres3} ]
[             {EOF   } ]

[ FSN        = {chiffres3} ]
[             {ANY   } ]
[             {NEXT  } ]

[ NBEFN     = {chiffres3} ]
[             {ALL   } ]

[ DENSITY    = {D1600  } ]
[             {D6250  } ];
[             {S35   } ]
[             {S75   } ];

```

"description-classe-d'appareils" se présente sous la forme :

```
DEVCLASS = classe-d'appareils
          { WORK
            { (nom-volume [nom-volume]...) }
            *
          }
```

"liste-identificateurs-d'appareils" se présente sous la forme :

```
DVIDLIST = (nom-appareil [nom-appareil]...)
          { WORK
            { (nom-volume [nom-volume]...) }
            *
          }
```

#### 4.3.4 Description

La première fonction de l'ordre ASSIGN est de faire connaître un fichier à une activité et de le lui rendre accessible. Pour ce faire, le nom de fichier interne (défini dans le programme) est rapproché du nom de fichier externe sous lequel le fichier est connu du système de gestion des fichiers.

La seconde fonction de l'ordre ASSIGN est d'allouer à l'activité les ressources (périphérique, support) associées au fichier. Les périphériques peuvent appartenir à un pool (voir l'ordre POOL).

Le fichier à affecter peut être une rubrique de description d'entrées (sous-fichier SYSIN, supprimé en fin de travail), un fichier permanent (catalogué ou non) ou un fichier temporaire. Il n'est possible d'accéder à un fichier catalogué que par l'intermédiaire du catalogue (voir l'ordre ATTACH). Pour un fichier catalogué, les paramètres d'implantation (RESIDENT, DEVCLASS/DVIDLIST et MEDIA) ne doivent pas être déclarés. Pour un fichier non catalogué ou temporaire, l'utilisateur doit au contraire les déclarer.

Si le nom de fichier externe commence par "SYS.", l'affectation se fait pour un fichier système. Les paramètres d'implantation ne doivent être déclarés ni dans ce cas, ni lorsque la valeur du deuxième paramètre positionnel commence par un astérisque (rubrique-de-description-d'entrées) ou est DUMMY.

La présence d'un ou plusieurs paramètres d'implantation (par exemple RESIDENT ou DEVCLASS et MEDIA) indique que le fichier est permanent et non catalogué, sauf si le paramètre TEMPY (ou FILESTAT=TEMPY) est également mentionné, auquel cas le fichier est considéré comme temporaire. La présence du paramètre CATALOG indique un fichier permanent catalogué.

Si le nom de fichier externe ne correspond ni à un fichier système, ni à une rubrique de description d'entrées, ni à un fichier fictif (DUMMY), et si les paramètres d'implantation FILESTAT, TEMPY et CATALOG sont absents de l'ordre ASSIGN :

- le système considère tout d'abord que le fichier est permanent et catalogué, et passe en revue tous les catalogues dans l'ordre suivant :
  1. les catalogues explicitement associés au travail par ATTACH,
  2. le catalogue de site (SITE.CATALOG),
  3. le catalogue système (SYS.CATALOG),
  4. les catalogues automatiquement associables enregistrés dans le système ;

- si le fichier est toujours introuvable, la gestion des fichiers considère qu'il s'agit d'un fichier permanent non catalogué, conservé d'une activité antérieure (voir le paramètre PASS ci-après) ;
- s'il n'existe pas de fichier non catalogué conservé d'une activité à une autre qui porte ce nom de fichier externe, le fichier est considéré comme résidant et non catalogué.

Pour faciliter l'identification, le système ajoute automatiquement une chaîne de caractères contenant le numéro de passage (RON) du travail en cours au début de tout nom de fichier temporaire. Même s'il se peut que ce nom composé figure dans le JOR, l'utilisateur ne doit employer que le nom de fichier temporaire simple.

Normalement, l'espace nécessaire à un fichier disque permanent doit faire l'objet d'une réservation statique au moyen des utilitaires de gestion de données (PREALLOC, FILALLOC, LIBALLOC, etc.). Toutefois, si un nouveau fichier disque est créé dans l'activité en cours, un ordre ALLOCATE doit figurer dans la rubrique de description d'activité correspondante. Cet ordre peut être omis pour un fichier temporaire (voir l'ordre ALLOCATE). Pour un fichier bande, aucune demande de réservation de place n'est nécessaire, bien que PREALLOC soit obligatoire si le fichier est catalogué, pour créer la rubrique de catalogue.

L'ordre ASSIGN permet aussi d'affecter directement à une activité un périphérique autre que disque ou bande. Si l'affectation concerne une imprimante, l'impression a lieu pendant l'exécution de l'activité ; le mécanisme SYSOUT n'est donc pas utilisé (voir l'ordre SYSOUT).

La gestion des fichiers ne reconnaît pas deux demandes simultanées relatives à un même volume lorsque l'une contient son nom d'appareil (DVIDLIST) et l'autre sa classe d'appareils (DEVCLASS).

- Remarques :**
1. Tous les fichiers système (ceux dont le nom commence par "SYS.") sont non catalogués et résidants ; il n'est donc pas nécessaire de spécifier leurs paramètres d'implantation (RESIDENT est la valeur implicite).
  2. Le système ajoute automatiquement un préfixe au début de tous les noms de fichiers externes commençant par un point (voir chapitre 3).

### Affectations multiples du même fichier dans une activité donnée

Plusieurs ordres ASSIGN concernant le même fichier permanent peuvent figurer dans une même activité, mais avec un nom de fichier interne différent pour chacun. Le nom de fichier externe doit au contraire être unique et les paramètres FILESTAT, LABEL, RESIDENT, DEVCLASS, MEDIA doivent être spécifiés (ou leur valeur implicite doit pouvoir s'appliquer).

S'ils sont spécifiés, les paramètres POOL, END et ABEND doivent avoir chaque fois une valeur identique.

L'option MOUNT ne doit pas être spécifiée en cas d'affectations multiples d'un même fichier dans une activité donnée, tous les volumes devant être montés.

Si les paramètres SHARE et ACCESS sont spécifiés, ils doivent être compatibles avec les autres spécifications des ordres ASSIGN dans lesquels ils apparaissent. Il en va de même pour le paramètre EXPDATE.

Lorsqu'une réservation de place est nécessaire pour le fichier, l'ordre ALLOCATE qui le concerne doit indiquer le nom de fichier interne correspondant du premier ordre ASSIGN de la description d'activité. La réservation de place ne devient effective qu'à l'ouverture du fichier sous ce nom de fichier interne. Enfin, s'il existe un ordre DEFINE, il ne s'applique qu'à un nom de fichier interne donné, et non au nom de fichier permanent correspondant.

### Concaténation de fichiers

La concaténation de fichiers permet de lire successivement plusieurs fichiers bande, disque ou disquette standard, ceux-ci étant traités par le programme comme un seul fichier logique séquentiel. Il suffit pour cela de spécifier les ordres ASSIGN dans l'ordre voulu, en indiquant un nom de fichier interne (nfi) uniquement dans le premier.

#### *Exemple :*

```
ASSIGN nfi, MY.FILEA, DEVCLASS=MT/T9, MEDIA=A1;
ASSIGN      , MY.FILEB, DEVCLASS=MT/T9, MEDIA=A2;
ASSIGN      , MY.FILEC, CATALOG=2;
```

Dans l'exemple ci-dessus, les trois fichiers sont considérés comme un seul fichier séquentiel commençant par MY.FILEA et se terminant par MY.FILEC (MY.FILEC étant un fichier bande catalogué).

### Restriction

Tous les fichiers à concaténer doivent être séquentiels. Il peut également exister des restrictions portant sur les différences d'attributs (telles que la longueur et le format des articles) des fichiers concernés. Ces restrictions sont fonction du langage de programmation (COBOL, FORTRAN) utilisé.

Si une concaténation concerne un fichier UFAS déclaré avec SHARE=MONITOR dans le catalogue ou dans l'ordre ASSIGN, il faut également spécifier, lorsque l'activité n'est pas relançable :

soit :

```
ACCESS = SPREAD (dans l'ordre ASSIGN)
```

ou

```
READLOCK = STAT (dans l'ordre DEFINE).
```

La concaténation des fichiers bande ou disquette non catalogués est demandée au moyen du paramètre NBEFN, qui indique le nombre de fichiers concernés en spécifiant le premier par son nom de fichier externe ou son numéro d'ordre (FSN).

Pour les bandes multifichiers, la concaténation s'effectue dans l'ordre physique d'enregistrement des fichiers sur la bande. Pour les disquettes, la concaténation s'effectue dans l'ordre d'apparition des labels dans le répertoire VTOC. Si la suite de fichiers disquette à concaténer comporte un fichier multifragment, le fichier traité immédiatement après est celui dont le label suit le label du dernier fragment. Une disquette ne pouvant comporter qu'un seul fragment de fichier, pluralité de fragments implique pluralité de volumes. Les fichiers dont le label apparaît, dans le répertoire de volume VTOC, entre le label du premier fragment et le label du dernier fragment d'un fichier disquette multifragment, ne sont pas pris en compte.

**Exemple 1 :**

```
ASSIGN INT1, FILE3, NBDFN=5, FSN=3,.....;
```

Concaténation de cinq fichiers d'une bande multifichier. La concaténation commence au troisième fichier du volume.

**Exemple 2 :**

```
ASSIGN INT1, FILE1, NBDFN=4,.....;
```

Concaténation de quatre fichiers disquette. Le premier fichier est désigné par son nom de fichier externe FILE1. Le second fichier est celui dont le label suit le label de FILE1 dans le répertoire de volume VTOC. Si le second fichier comporte trois fragments, la concaténation s'effectue comme suit :

	1	2	3	4	5
Volume disquette 1	XYZ	FILE1	FILE2	ABC	AXY
			fragm.1		
Volume disquette 2	ACX	FILE2	FILE9	RST	
			fragm.2		
Volume disquette 3	FILE2	FILE3	FILE4	FILE5	
			fragm.3		

Lorsque le traitement des trois fragments du fichier FILE2 est terminé, les fichiers FILE3 et FILE4 sont concaténés. Les fichiers ABC, AXY, ACX, FILE9, RST et FILE5 ne sont pas concaténés.

**Remarques sur les suffixes de noms de fichiers internes**

Lorsqu'un programme COBOL accède à certains types de fichiers indiqués ci-après, le nom de fichier interne de l'ordre ASSIGN approprié doit comporter un suffixe, qui se présente sous la forme d'un trait d'union suivi d'un ou deux chiffres. Le nom de fichier interne composé ne doit pas dépasser la longueur maximum normale des noms de fichiers internes (huit caractères). Pour plus de précisions sur les bandes multifichiers et les fichiers à multiples unités logiques, voir le guide utilisateur COBOL (06UL).

Les règles d'écriture des noms de fichiers internes figurant dans les ordres ASSIGN sont les suivantes :

- Pour un fichier sur bande multifichier, ce nom doit comporter le nom de fichier interne COBOL suivi d'un suffixe indiquant le numéro d'ordre du fichier sur la bande.

**Exemples :**

```
ASSIGN SUBFIL-1, CA.MTAPE, DEVCLASS=MT/T9, MEDIA=V66, FSN=1;
```

```
ASSIGN SUBFIL-2, CA.MTAPE, DEVCLASS=MT/T9, MEDIA=V66, FSN=2;
```

```
ASSIGN SUBFIL-6, CA.MTAPE, DEVCLASS=MT/T9, MEDIA=V66, FSN=6;
```

SUBFIL-14 n'est pas un nom de fichier interne valable, car il comporte plus de huit caractères.

- Pour une unité logique, ce nom doit comporter le nom de fichier interne COBOL suivi d'un suffixe indiquant le numéro d'ordre de l'unité logique.

**Exemples :**

```
ASSIGN MLNFIL-1, MY.FILEA;
```

```
ASSIGN MLNF1L-2, MY.FILEB;
```

### Commentaires et restrictions

Si le programme COBOL contient une instruction ACCEPT FROM SYSIN (voir le manuel de référence COBOL - 05UL), l'ordre ASSIGN doit avoir la forme suivante :

```
ASSIGN H_RD, *nom-rubrique-de-description-d'entrées;
```

Le nombre total d'instructions ASSIGN, DEFINE et ALLOCATE pouvant figurer dans une rubrique de description d'activité ne doit pas dépasser 1200. Ce maximum prend en compte les affectations multiples d'un même fichier, les affectations de fichiers concaténés, ainsi que les fichiers affectés par un ordre MWLINLB ou MWLIB en environnement interactif (voir les commandes GCL MWINLIB et MWLIB).

### 4.3.5 Paramètres

#### 4.3.5.1 Paramètres obligatoires

##### **nom-fichier-interne**

C'est le nom déclaré dans le programme origine. Il ne doit pas dépasser huit caractères (lettres, chiffres, trait d'union et caractère de soulignement).

Il s'agit du premier paramètre positionnel de l'ordre ASSIGN. Il ne peut être omis qu'en cas de concaténation de fichiers (voir plus haut).

##### **Exemples :**

INFILE

INF4

MY-OUT

##### **nom-fichier-externe**

C'est le nom d'un fichier permanent sous lequel celui-ci est connu du système. Les règles de formation du nom de fichier externe sont détaillées au chapitre 3 de ce manuel.

Il s'agit d'un paramètre positionnel, qui peut être omis si DEVCLASS est une imprimante (cas d'une affectation directe de périphérique autre que disque ou bande - voir le manuel *Unit Record Devices User's Guide*).

##### **nom-rubrique-de-description-d'entrées**

Paramètre positionnel identifiant les données de la rubrique de description d'entrées à affecter à l'activité. Ces données sont enregistrées dans un fichier SYSIN et supprimées en fin de travail. "nom-rubrique-de-description-d'entrées" ne doit pas dépasser 16 caractères (lettres, chiffres, trait d'union et caractère de soulignement) et doit toujours être précédé d'un astérisque (\*). Lorsqu'il est spécifié, le seul autre paramètre valable est "nom-fichier-interne".

##### **nom-fichier-temporaire**

Paramètre positionnel identifiant un fichier temporaire. "nom-fichier-temporaire" ne doit pas dépasser 16 caractères (lettres, chiffres, trait d'union et souligné) et doit commencer par une lettre ou un chiffre. Le paramètre TEMPRLY ou FILESTAT=TEMPRLY doit toujours être spécifié pour un fichier temporaire. Noter que la longueur maximum d'un nom de fichier temporaire de format ANSI sur bande est de 9 caractères.

**DUMMY**

Paramètre positionnel indiquant qu'aucun volume n'est nécessaire pour contenir le fichier. Aucune opération portant sur ce fichier n'est prise en compte, à l'exception de la lecture, qui provoque une condition fin de fichier. Dans certains langages de programmation (le COBOL, par exemple), DUMMY est limité aux fichiers d'entrée séquentiels (voir le guide utilisateur correspondant).

\*

Ce paramètre n'est utilisable qu'avec les bandes et les disquettes non cataloguées. Il permet d'affecter un fichier sans spécifier son nom de fichier externe.

Pour les bandes, il représente soit le fichier dont la position sur la bande est donnée par le numéro d'ordre du fichier (FSN), soit le premier fichier de la bande si le numéro d'ordre de fichier est omis.

Pour les disquettes, il représente le fichier dont le label figure en tête du répertoire de volume VTOC.

**Restriction**

Un fichier spécifié sous la forme nfe=\* ne peut ni être conservé d'une activité à une autre, ni faire l'objet d'une relance sur point de reprise. Toute tentative provoque un arrêt prématuré et le code retour CONFLICT est émis à la relance.

## 4.3.5.2 Paramètres facultatifs

**FILESTAT**

FILESTAT peut servir d'élément d'identification d'un fichier (voir chapitre 3).

Il indique l'état du fichier, correspondant à l'une des valeurs suivantes :

CAT	Fichier permanent catalogué. Si aucun paramètre d'implantation (RESIDENT, VOLSET, DEVCLASS/DVIDLIST et MEDIA) n'est spécifié, c'est la valeur implicite de FILESTAT.
UNCAT	Fichier permanent non catalogué. Si FILESTAT n'est pas spécifié mais que RESIDENT, ou DEVCLASS/DVIDLIST l'est, FILESTAT=UNCAT est implicite.
TEMPRY	Fichier temporaire. FILESTAT peut être omis dans ce cas (voir ci-après).

**Remarque :**

Etant donné qu'il existe des valeurs implicites de FILESTAT pour les fichiers catalogués et non catalogués, et que le paramètre autodéfini TEMPY peut être utilisé pour les fichiers temporaires (voir ci-après), le paramètre FILESTAT peut généralement être omis de l'ordre ASSIGN.

Il doit cependant être spécifié dans les cas où son absence risquerait de créer une ambiguïté : par exemple, dans un ordre ASSIGN relatif à un fichier conservé d'une activité à une autre (pour lequel les paramètres d'implantation ne sont pas nécessaires - voir PASS ci-après).

### **TEMPRY**

La spécification de ce paramètre équivaut à celle de FILESTAT=TEMPRY. Il faut spécifier l'un de ces deux paramètres pour affecter un fichier temporaire. Si aucun paramètre d'implantation n'est indiqué et que le fichier ne provient pas d'une activité antérieure (paramètre PASS), ce fichier est implanté sur le Volset par défaut du soumetteur du travail, à condition que la fonction VOLSET soit active. Dans le cas contraire, le fichier est implanté sur le volume résident.

### **CATALOG**

Ce paramètre s'applique uniquement aux fichiers catalogués (si CATNOW n'est pas précisé). Il sert à identifier un catalogue donné parmi ceux associés au travail par l'ordre ATTACH, dans le but de trouver le fichier spécifié (voir la description de l'ordre ATTACH dans le *Guide utilisateur de gestion des catalogues*). La valeur du paramètre CATALOG représente la position du catalogue voulu dans l'ordre ATTACH.

Si le système ne trouve pas la rubrique correspondant au fichier dans le catalogue indiqué, l'activité est arrêtée prématurément.

Si le paramètre CATALOG est omis, tous les catalogues sont passés en revue dans l'ordre où ils sont spécifiés (catalogues associés au travail par ATTACH, catalogue de site, catalogue système et catalogues automatiquement associables) jusqu'à ce que la rubrique correspondant au fichier soit trouvée.

Si CATNOW est également spécifié, CATALOG identifie le catalogue dans lequel le fichier doit être catalogué.

### **CATNOW**

Crée dynamiquement une rubrique de catalogue pour le fichier affecté. Pour désigner le catalogue concerné, il est possible de le spécifier explicitement (CATALOG=chiffre1) ou de passer tous les catalogues en revue.

CATNOW doit être spécifié si le fichier est à cataloguer. Le paramètre d'implantation doit également être spécifié à l'aide du paramètre RESIDENT (sur un disque résident), d'une identification de Volset, ou d'un descripteur de classe d'appareils (une liste d'identificateurs d'appareils n'est pas autorisée).

Si le paramètre CATNOW est utilisé, un ordre ALLOCATE doit être associé à l'ordre ASSIGN, sauf pour les fichiers bande.

Pour plus de précisions sur le paramètre CATNOW, voir le *Guide utilisateur de gestion des catalogues*.

**EXPDATE**

Ce paramètre indique la date de péremption d'un fichier. Dans le cas d'un fichier bande, EXPDATE est une protection contre les tentatives accidentelles d'écrasement du nom de fichier. Dans le cas d'un fichier disque pour lequel il existe une réservation de place par l'ordre ALLOCATE dans l'activité en cours, EXPDATE empêche l'écrasement accidentel du nom de fichier et la libération involontaire de la place réservée au fichier.

**Remarques :**

1. La date de péremption d'un fichier disque est spécifiée soit lors de la réservation de place au moyen de l'utilitaire PREALLOC (voir chapitre 5), soit dans l'ordre ASSIGN auquel est associé un ordre ALLOCATE, lorsque le fichier est utilisé pour la première fois. Cette date est enregistrée dans le label du fichier.
2. L'utilitaire DEALLOC permet de libérer la place affectée à un fichier disque, c'est-à-dire de supprimer ce fichier (voir chapitre 5). Tant que la date de péremption n'est pas atteinte, la place réservée au fichier ne peut être libérée que si le paramètre BYPASS est spécifié dans l'ordre DEALLOC.
3. Un fichier disque peut être ouvert en mode sortie et son contenu antérieur écrasé, même s'il n'a pas atteint la date de péremption indiquée. Autrement dit, le nom du fichier est protégé, mais pas son contenu.
4. Le paramètre EXPDATE n'a de sens dans l'ordre ASSIGN pour les fichiers disque non catalogués que si la réservation de place pour le fichier (par l'ordre ALLOCATE) figure dans l'activité en cours.
5. Le paramètre EXPDATE n'a de sens pour les fichiers catalogués (disque et bande) que si une nouvelle génération est créée. Dans ce cas, la valeur indiquée prévaut sur la date de péremption enregistrée dans le catalogue.
6. Il est possible d'attribuer une nouvelle date de péremption à un fichier bande non catalogué chaque fois qu'une écriture a lieu dans ce fichier (autrement dit, lorsqu'il est ouvert en mode sortie). Si aucune date n'est spécifiée, la date du jour est implicitement retenue.
7. Lorsqu'une date de péremption est spécifiée pour un fichier temporaire, elle n'est pas prise en compte.
8. L'utilitaire VOLPREP peut détruire tous les fichiers existant sur le disque traité. Pour que la suppression ait lieu, il faut mentionner BYPASS ; s'il existe un fichier dont la date de péremption n'est pas atteinte, un message console demande alors à l'opérateur si tous les fichiers doivent être supprimés, quelle que soit leur date de péremption.

9. Tant que la date de péremption d'un fichier bande n'est pas atteinte, le système le protège contre les risques d'écrasement par un fichier de nom différent ; pour qu'un nouveau fichier de nom différent puisse être écrit sur la bande, il faut d'abord la réinitialiser au moyen de l'utilitaire VOLPREP (voir chapitre 5). L'existence d'une période de validité n'empêche pas le fichier d'être ouvert en mode sortie et rechargé, mais le nom de fichier utilisé doit alors être identique à celui qui figure dans le label du fichier.

La date de péremption d'un fichier peut être spécifiée sous l'une des trois formes suivantes :

- En nombre de jours : par exemple, 12.
- En précisant l'année (sur deux chiffres) et le numéro du jour dans l'année (sur trois chiffres) : par exemple, 91/046 signifie le 46e jour de l'année 1991, soit le 15 Février 1991.
- En précisant l'année (sur deux chiffres), le mois (sur deux chiffres) et le jour (sur deux chiffres) : par exemple, 91/02/15 signifie le 15 Février 1991.

#### **VOLWR (volumes bande et cartouche seulement)**

La bande ou la cartouche doit être montée sur le volume avec le mécanisme de protection en écriture sur OFF (avec anneau d'autorisation d'écriture s'il s'agit d'une bande, en position d'autorisation d'écriture dans le cas d'une cartouche).

#### **NVOLWR (volumes bande seulement)**

La bande ou la cartouche doit être montée avec le mécanisme de protection sur ON (sans anneau d'autorisation d'écriture s'il s'agit d'une bande, en position d'interdiction d'écriture dans le cas d'une cartouche).

#### **OPTIONAL**

Le fichier est considéré comme fictif (DUMMY) si l'opérateur indique au système que les volumes du fichier ne sont pas disponibles, ou si les volumes sont disponibles mais ne contiennent pas effectivement le fichier.

#### **DEFER**

L'activité peut commencer même si le volume indiqué n'est pas monté. Dans ce cas, le volume est monté au moment de l'ouverture du fichier associé au nom de fichier interne. Un appareil est cependant spécifié, afin que le volume puisse être monté à l'ouverture du fichier ; en effet, si aucun appareil ne se trouvait disponible à ce moment, l'exécution du travail serait arrêtée prématûrement. Entre l'affectation et l'ouverture du fichier, le message suivant est affiché à la console, sans nom de volume :

MSxx ALLOCATED TO RON

## **RESIDENT**

Indique que le fichier est enregistré sur des volumes disque résidants.

## **DEVCLASS (description-classe d'appareils)**

Indique la classe d'appareils (voir chapitre 3) du périphérique associé à un fichier non catalogué, ou à un fichier (temporaire ou à cataloguer si CATNOW est spécifié) devant être affecté automatiquement. Tous les appareils de cette classe sont équivalents ; l'appareil utilisé est déterminé par GCOS.

## **DVIDLIST (liste-identificateurs-d'appareils)**

Spécifie par leur nom un appareil ou une liste d'appareils (voir chapitre 3) sur lesquels les volumes doivent être montés. Pour un fichier non catalogué, il est préférable d'utiliser le paramètre DEVCLASS plutôt que le paramètre DVIDLIST.

- Remarques :**
1. Les volumes disque résidants ne peuvent pas être spécifiés par leur nom d'appareil.
  2. Les fichiers affectés par un ordre ASSIGN comportant un paramètre DVIDLIST ne sont pas partageables.

## **MEDIA**

Ce paramètre est spécifié avec DEVCLASS ou DVIDLIST. Il spécifie les noms des volumes non résidants qui contiennent le fichier (non catalogué, temporaire ou en passe d'être affecté et catalogué si CATNOW est indiqué). Un nom de volume peut comporter jusqu'à six caractères alphanumériques.

Au moment de l'exécution du programme, un message demandant le montage du ou des volumes nécessaires s'affiche à la console opérateur, sauf si le ou les volumes contenant le fichier sont résidants ou ont été montés à l'avance.

Pour une imprimante, le nom de volume se décompose comme suit :

- caractères 1 et 2 : identificateur de bande d'impression ;
- caractères 3 à 6 : numéro de type de feuillet.

### **WORK**

Cette valeur ne peut être spécifiée que pour les bandes de sortie. Elle demande l'utilisation d'un certain nombre (valeur du paramètre MOUNT) de bandes de manoeuvre avec labels standard (NATIVE).

\*

Cette valeur signifie que les noms de volumes ne sont pas connus à la soumission du travail, mais seront fournis par l'opérateur au moment de l'exécution. Elle s'applique aux fichiers bande, catalogués ou non, et aux fichiers disquette, en entrée comme en sortie.

MEDIA=\* interdit la reconfiguration des appareils et le prémontage des volumes. Les volumes utilisés et leur ordre de montage étant sous le seul contrôle de l'opérateur, il faut s'assurer qu'il possède une liste des volumes à monter dans le bon ordre, et qu'il peut facilement identifier chaque volume.

**Remarques :**

1. Un fichier sur bande de manoeuvre peut être utilisé dans plusieurs activités d'un travail si END=PASS figure dans chacune des activités qui l'utilisent.
2. Il est possible d'interdire l'accès d'une bande de manoeuvre aux autres utilisateurs en spécifiant END=PASS dans l'ordre ASSIGN pour chaque activité devant accéder à un fichier enregistré sur cette bande.
3. L'utilitaire VOLPREP ou TAPEPREP peut servir à préparer une bande de manoeuvre.
4. Si TEMPRY ou FILESTAT=TEMPRY n'est pas spécifié dans ASSIGN lorsqu'un programme écrit dans un fichier sur bande de manoeuvre, cette dernière cesse d'être considérée comme une bande de manoeuvre.
5. Si les volumes spécifiés (par exemple dans MEDIA) pour un fichier bande ordinaire ouvert en sortie ne suffisent pas à contenir le fichier, le système lui affecte automatiquement des bandes de manoeuvre (le nombre total de 10 volumes ne pouvant être dépassé).

## **VOLSET**

Ce paramètre, qui spécifie le nom de Volset dans lequel doit résider le fichier, ne peut être utilisé que lorsque la fonction VOLSET est active sur le système et que le fichier est temporaire ou à cataloguer (CATNOW est également spécifié). VOLSET s'applique aux fichiers inexistants et aux fichiers préalablement affectés (par exemple, un fichier temporaire conservé d'une activité antérieure à l'aide du paramètre PASS). Il n'est pas compatible avec les autres paramètres d'implantation (RESIDENT, DEVCLASS/DVIDLIST et MEDIA). Le nom de Volset consiste en six caractères alphanumériques maximum, choisis par l'administrateur du système.

DFLT est un nom symbolique réservé à la dénomination du Volset par défaut :

- dans la fonction de base, le Volset de site constitue le Volset par défaut de tous les projets (il s'agit du seul Volset disponible) ;
- dans la fonction étendue, l'administrateur du système définit un Volset par défaut pour le projet ; si un projet n'est pas associé à un Volset par défaut, il adopte implicitement le Volset de site.

Pour plus d'informations, voir le *Guide d'administration du gestionnaire d'espace disque*.

**SUBFILE**

Ce paramètre concerne uniquement les unités de bibliothèque (la bibliothèque est identifiée par "nom-fichier-externe"). Il permet d'accéder séquentiellement à l'unité spécifiée.

Le nom d'unité peut comporter jusqu'à 31 caractères (lettres, chiffres, trait d'union et caractère de soulignement). Il doit commencer par une lettre ou un chiffre.

Il peut s'agir par exemple d'une unité de bibliothèque SL contenant la description d'un réseau de communication, ou d'une unité contenant des sorties envoyées à SYSOUT.

**SHARE**

Ce paramètre indique les règles de partage d'un fichier à l'intérieur d'une même activité (cas des affectations multiples) ou entre des activités concurrentes. Lorsque SHARE n'est pas spécifié, ce sont ses valeurs implicites qui s'appliquent (voir remarque ci-après).

NORMAL	Un accès en écriture (processeur de sortie) ou plusieurs accès en lecture (processeurs d'entrée).
	Deux ou plusieurs affectations simultanées du même fichier avec des valeurs de SHARE différentes sont interdites, sauf avec SHARE=NORMAL et SHARE=ONEWRITE. Si par exemple un travail est exécuté avec les paramètres SHARE=FREE et ACCESS=WRITE (voir ci-dessous le paramètre ACCESS), une demande spécifiant SHARE=ONEWRITE et ACCESS=READ ne sera pas satisfaite tant que le premier travail n'aura pas libéré le fichier. Noter qu'après une affectation avec SHARE=NORMAL et ACCESS=READ, une demande avec SHARE=ONEWRITE et ACCESS=WRITE n'est pas satisfaite.
ONEWRITE	Un accès en écriture et plusieurs accès en lecture.
FREE	Aucune restriction de partage.
MONITOR	Cette valeur s'applique uniquement aux fichiers disque UFAS et IDS/II. Elle indique que le partage du fichier est contrôlé par GAC (gestion générale des accès). Le fichier peut être affecté simultanément à plusieurs activités en lecture et en écriture. Le partage des fichiers est alors contrôlé au moment de l'accès au niveau de l'intervalle de contrôle (CI), au lieu d'être contrôlé à l'affectation au niveau du fichier.
DIR	Cette valeur s'applique uniquement aux bibliothèques. Elle indique que chaque unité peut être utilisée par plusieurs lecteurs ou par un programme d'écriture.

**Remarque :** Pour les fichiers non catalogués, la valeur implicite de SHARE est NORMAL, sauf lorsque le paramètre SUBFILE est spécifié, auquel cas elle est DIR (s'il s'agit d'une bibliothèque).

Pour les fichiers catalogués, le mode implicite de partage est celui indiqué dans la rubrique de catalogue correspondante. Si une valeur de SHARE différente de celle de la rubrique est spécifiée, c'est celle de la rubrique qui prévaut, et l'activité a l'accès exclusif au fichier.

## ACCESS

Indique les modes de traitement autorisés pour l'activité.

WRITE	Autorise les modes entrée, sortie et adjonction en fin de fichier.
READ	Autorise le mode entrée seulement.
SPWRITE	Modes entrée, sortie et adjonction en fin de fichier comme pour WRITE. De plus, l'activité en cours a seule accès au fichier, quelle que soit la valeur de SHARE.
SPREAD	Mode entrée seulement, comme pour READ. De plus, l'activité en cours a seule accès au fichier, quelle que soit la valeur de SHARE.
RECOVERY	S'applique uniquement aux fichiers catalogués (voir le guide utilisateur de gestion des catalogues - 35UF). L'activité a seule accès au fichier pour la reconstitution du fichier.
ALLREAD	Mode entrée, comme pour READ. De plus, les autres activités ne peuvent avoir accès au fichier qu'en mode READ, quelle que soit la valeur de SHARE.

**Remarque :** Lorsqu'un ordre ALLOCATE est associé au fichier, ce dernier n'est accessible qu'à l'activité (ACCESS=READ est donc traité comme ACCESS=SPREAD et ACCESS=WRITE est traité comme ACCESS=SPWRITE).

## END

Ce paramètre indique au système l'état dans lequel doit être laissé le fichier en fin normale d'activité :

PASS	Si le fichier est temporaire, il est conservé en vue de son utilisation par une activité ultérieure du même travail soit jusqu'à la fin du travail, soit jusqu'à ce qu'une activité ultérieure l'affecte sans option PASS, auquel cas il est supprimé à la fin de cette activité. Si le fichier est permanent, l'identification de volume et d'appareil associée au fichier est conservée. Il est donc inutile, pour un fichier temporaire ou permanent affecté à une activité ultérieure du travail, de mentionner les paramètres DEVCLASS (ou DVIDLST),
------	---

MEDIA, RESIDENT, VOLSET et LABEL dans l'ordre ASSIGN correspondant.

Pour qu'une activité puisse accéder à un fichier conservé d'une activité précédente, elle doit comporter un ordre ASSIGN spécifiant un nom de fichier interne, un nom de fichier externe et le paramètre FILESTAT ou TEMPRY ; s'il spécifie également les paramètres DEVCLASS, (ou DVIDLIST), MEDIA, RESIDENT, VOLSET ou LABEL, leurs valeurs doivent concorder avec celles spécifiées dans l'activité d'origine.

Les valeurs éventuellement spécifiées pour MOUNT, END, ABEND, SHARE et ACCESS ne s'appliquent qu'à l'activité dans laquelle elles sont spécifiées (elles ne sont pas automatiquement conservées d'une activité à une autre).

**Remarque :** Un fichier dont le nom de fichier externe est \* ne peut pas être conservé d'une activité à une autre.

#### DEASSIGN

Un fichier temporaire est supprimé en fin d'activité ou lorsqu'une instruction CLOSE WITH LOCK est exécutée dans le programme (COBOL). Dans le cas d'un fichier permanent non catalogué affecté à une activité ultérieure d'un travail, l'ordre ASSIGN suivant doit indiquer l'identification complète du volume et de l'appareil (RESIDENT, ou DEVCLASS/DVIDLIST et MEDIA).

#### LEAVE

S'applique uniquement aux fichiers bande ou cartouche, avec les résultats suivants :

A la fermeture du fichier, le support reste dans la position où il se trouve.

Au moment du changement de volume, le support est rembobiné, sauf si :

- l'ordre ASSIGN spécifie MD=\* (n'importe quel support), auquel cas le support est déchargé ;
- l'ordre ASSIGN spécifie FSN=NEXT et l'accès est en mode INPUT (par exemple, pour ACCESS=READ, ou INSET avec les utilitaires), auquel cas le support reste dans sa position en cours (EOT) ;
- le fichier est un fichier cartouche implanté sur un dérouleur de bandes doté d'un chargeur de cartouches SCL (Sequential Cartridge Loader) en mode automatique, auquel cas le support est déchargé ;
- le programme comporte une demande LEAVE dans la primitive de changement de volume.

En fin d'activité, le dernier support traité reste dans sa position en cours.

Dans tous les cas, le volume reste affecté jusqu'à la fin de l'activité, même si la bande ou l'appareil est libéré.

**ATTENTION**

LEAVE présente un intérêt lorsque le nombre d'unités de bande est supérieur ou égal au nombre de volumes cités dans MEDIA.

**UNLOAD**

S'applique uniquement aux volumes bande ou cartouche, avec les résultats suivants :

Au moment du changement de volume, le support est déchargé, sauf si :

- a) le paramètres ENDVOL est spécifié
- b) les fichiers sont traités par des utilitaires en tant qu'INSET, auquel cas le support reste dans sa position en cours (EOT) ;
- c) les fichiers sont traités par des utilitaires en tant qu'OUTSET, auquel cas le support est rembobiné ;
- d) les fichiers sont des fichiers cartouche implantés sur un dérouleur de bandes doté d'un chargeur de cartouche SCL (Sequential Cartridge Loader), en mode manuel, auquel cas b) ou c) s'applique, selon le type de traitement.

**Remarques :**

- 1. Si, en mode automatique, l'option END=UNLOAD est spécifiée pour des fichiers sur cartouche chargés sur un dérouleur comportant le chargeur de cartouches SCL (Sequential Cartridge Loader), la dernière cartouche utilisée, déclarée dans la liste des supports de l'ordre ASSIGN correspondant, est déchargée lorsqu'elle cesse d'être affectée à l'activité, ainsi que les suivantes qui ne sont pas utilisées, s'il y en a. En mode manuel, seule la dernière cartouche utilisée est déchargée.
- 2. END ne peut être spécifié si \*nom-rubrique-de-description-d'entrées l'est.
- 3. Un fichier temporaire multivolume pour lequel MOUNT a été spécifié ne peut être totalement supprimé que s'il y a un nombre suffisant d'appareils disponibles pour monter tous les volumes du fichier en même temps.

Si MOUNT n'est pas spécifié (autrement dit, si tous les volumes doivent implicitement être montés) mais que END=DEASSIGN l'est, la suppression totale du fichier est garantie.

Si MOUNT n'est pas spécifié mais que END=PASS l'est, le fichier n'est totalement supprimé que s'il y a un nombre suffisant d'appareils disponibles pour monter tous les volumes en même temps.

Si END=PASS est spécifié, la suppression a lieu au prochain ordre ASSIGN avec END=DEASSIGN, ou au plus tard, en fin de travail.

4. Les bandes non natives (traitées par le processeur FTP) sont toujours considérées comme des volumes monofichiers par GCOS 7.
5. Un fichier bande pour lequel LEAVE ou UNLOAD n'a pas été déclaré ou pour lequel PASS a été spécifié est implicitement rembobiné, avec les restrictions suivantes :
  - lors de la fermeture, la bande est rembobinée lorsque l'affectation (ASSIGN) a été faite avec l'option POOL, ou lorsque la concaténation de fichiers a été déclarée ;
  - en cas de bascule de volume, voir END=LEAVE ;
  - en fin d'activité, le volume bande est rembobiné, sauf si l'affectation (ASSIGN) a été faite avec MD=\*.

### ATTENTION

Lorsqu'une bibliothèque de cartouches est utilisée, il est possible de forcer le démontage. Reportez-vous au paragraphe Démontage d'un volume de bibliothèque du manuel intitulé *Cartridge Tape Library Unix Server User's Guide*.

### ABEND

Ce paramètre indique au système l'état dans lequel doit être laissé le fichier en cas d'arrêt anormal d'activité. Les valeurs PASS, DEASSIGN, LEAVE et UNLOAD qu'il peut prendre sont décrites dans le paramètre END ci-dessus.

### ENDVOL

S'applique à un lecteur de bande ou de cartouche avec un fichier multi-volume, ce paramètre concerne tous les volumes sur lesquels le fichier s'étend sauf le dernier qui est traité par le paramètre END. Il indique au système l'état dans lequel le volume doit être laissé, soit :

UNLOAD	les volumes intermédiaires sont déchargé lors du changement de volume.
REWIND	les volumes intermédiaires sont rembobinés lors du changement de volume.
LEAVE	les volumes intermédiaires sont laissés dans leur position courante lors du changement de volume.

## MOUNT

Le paramètre MOUNT s'applique aux fichiers UFAS séquentiels.

Ce paramètre définit le nombre d'appareils affectés au fichier pour l'activité, ainsi que le nombre de volumes d'une classe d'appareils donnée devant être montés au lancement de l'activité (avant que l'exécution de l'activité commence) lorsque le paramètre POOL n'est pas spécifié (voir ci-après la description du paramètre POOL). Les noms de volumes figurent dans le paramètre MEDIA ou sont sélectionnés par la fonction VOLSET si le paramètre VOLSET est utilisé. Si les volumes ne sont pas tous montés au moment du lancement de l'activité, les volumes restants sont montés en cours d'exécution de l'activité, en fonction des besoins du programme. Si MOUNT n'est pas spécifié, tous les volumes définis par le paramètre MEDIA sont montés implicitement lorsqu'il s'agit de disques. La valeur implicite est MOUNT=1 pour tous les types de périphériques sauf l'unité de disques.

**Remarque :** Si MEDIA=WORK est spécifié, MOUNT indique le nombre d'appareils à affecter.

## POOL

Ce paramètre indique que les volumes qui contiennent le fichier doivent être montés (voir le paramètre MOUNT) sur des appareils de pool ayant les mêmes attributs.

Ces appareils peuvent être affectés à un pool par un ou plusieurs ordres POOL. Le ou les ordres POOL correspondants doivent précéder l'ordre ASSIGN dans la rubrique de description de l'activité.

Lorsque le paramètre POOL est spécifié dans un ordre ASSIGN, l'ordre POOL est facultatif. Si l'ordre POOL est omis, le pool d'appareils est affecté dynamiquement et selon les valeurs des paramètres correspondants de l'activité.

Lorsque POOL est spécifié, un appareil de pool est réservé pour utilisation avant l'exécution de l'activité et affecté au fichier au moment de son ouverture. Cet appareil n'est libéré et restitué au pool de périphériques qu'au moment de la fermeture du fichier (en COBOL par exemple, au moyen de l'option CLOSE WITH LOCK ou CLOSE). L'utilisation des pools de périphériques est décrite en détail dans le *Guide utilisateur JCL*.

**FIRST** Cette valeur ne peut être spécifiée que dans le paramètre POOL. Elle signifie que le ou les volumes (selon la valeur du paramètre MOUNT) contenant le fichier cité dans cet ordre ASSIGN de l'activité doivent être montés sur un ou plusieurs appareils de pool, avant l'ouverture du fichier associé au nom de fichier interne dans l'activité et avant que l'exécution de l'activité ne commence. Il en résulte que la somme de tous les appareils spécifiés avec POOL=FISRT ne doit pas dépasser le nombre minimum d'appareils définis dans l'ordre POOL correspondant.

**NEXT** Cette valeur ne peut être spécifiée que dans le paramètre POOL. Elle signifie que les volumes contenant le fichier cité dans cet ordre ASSIGN de l'activité doivent être montés sur des appareils de pool pendant l'exécution de l'activité, en fonction des besoins du programme.

## LABEL

Ce paramètre indique le type des labels de volume. Il peut prendre l'une des valeurs suivantes :

NATIVE	Labels GCOS standard.
NONE	Absence de label.
NSTD	Labels de format inconnu.

Le système considère implicitement que les volumes bande et disque ont des labels standard (LABEL=NATIVE). Si un volume avec labels non standard est soumis au système, l'utilisateur doit indiquer explicitement le type des labels. Si la valeur de LABEL (NATIVE implicitement) ne correspond pas au type de labels rencontré par la fonction GCOS de reconnaissance automatique de volume (AVR), l'activité n'est pas lancée.

La valeur NONE indique qu'il n'y a pas de labels reconnaissables sur le volume. Lorsque LABEL=NONE est spécifié pour un volume bande d'entrée, les méthodes d'accès de GCOS interprètent la première marque de bande rencontrée comme une fin de fichier.

## FIRSTVOL

Ce paramètre s'applique aux fichiers multivolumes. Il définit le numéro ordinal du premier volume auquel accéder à l'ouverture du fichier, c'est-à-dire la position dans la liste des supports du premier volume à monter. Pour les fichiers catalogués, cette liste est enregistrée dans le catalogue ; pour les fichiers non catalogués, elle est spécifiée dans le paramètre MEDIA de l'ordre ASSIGN.

Il ne peut pas être spécifié si le fichier n'est pas catalogué et si le mode de traitement n'est ni APPEND ni INPUT. Si FIRSTVOL est spécifié pour un fichier avec labels, la gestion des fichiers vérifie que le numéro d'ordre du volume choisi (par rapport au début du fichier) est bien identique à la valeur de FIRSTVOL. Si FIRSTVOL n'est pas spécifié, l'activité se poursuit sans aucun contrôle de numéro.

FIRSTVOL=EOF signifie que le premier volume à traiter est celui qui contient la marque fin de fichier. Cette valeur s'applique uniquement aux fichiers catalogués.

### *Exemple :*

Soit une liste de supports (V1, V2, V3, V4). Si FIRSTVOL=3, le système demandera le montage de V3, et au moment de l'ouverture du fichier, contrôlera le numéro d'immatriculation du volume V3 (pour vérifier qu'il s'agit bien du troisième volume du fichier).

## LASTVOL

Ce paramètre s'applique aux fichiers multivolumes. Il définit le numéro ordinal du dernier volume auquel accéder, c'est-à-dire la position dans la liste de supports du volume contenant la fin du fichier. La fin des données (code retour DATAFLIM) est signalée lorsque le traitement de ce volume est terminé.

LASTVOL=EOF signifie que le dernier volume à traiter doit contenir la marque fin de fichier. Pour un fichier non catalogué, une erreur est donc signalée au moment de l'accès au dernier article du volume, si la liste de supports est incomplète.

LASTVOL=EOF est la valeur implicite appliquée aux fichiers séquentiels, seuls fichiers pour lesquels il n'est pas nécessaire que tous les volumes (bandes et disquettes) soient montés au lancement de l'activité.

LASTVOL ne peut pas être spécifié pour un fichier disque si une demande d'affectation a déjà été émise.

**Exemple :**

```
ASSIGN LFILE, MY.BFIL, LASTVOL=EOF DEVCLASS=MT/T9/D1600,  
MEDIA=(V1,V2);
```

Dans l'exemple ci-dessus, si V2 ne contient pas la fin du fichier MY.BFIL, l'activité sera arrêtée prématurément.

LASTVOL=EOF étant la valeur implicite pour les bandes, elle aurait pu être omise.

Si LASTVOL=2 était spécifié, l'activité se terminerait normalement, même si V2 ne contient pas la marque de fin de fichier.

**FSN**

Ce paramètre s'applique uniquement aux volumes bande multifichiers. Il est obligatoire pour tous les fichiers présents sur une bande ou plus. Il définit le numéro d'ordre du fichier (sa position relative) dans la famille de fichiers.

Lorsque FSN=chiffres3, la valeur spécifiée doit être un nombre entier compris entre 1 et 253 inclus.

Si FSN=ANY, la gestion de fichiers recherchera le fichier sur la bande au moment de son ouverture.

Si FSN=NEXT pour un fichier en mode sortie, le nouveau fichier suivra les fichiers existant sur la bande, même si un fichier de même nom y figure déjà (lorsque FSN=ANY, le fichier de même nom est écrasé).

**Remarques :**

1. En sortie, un fichier identifié par FSN peut être écrasé par un autre fichier (à condition que sa date de péremption soit atteinte - voir ci-dessus le paramètre EXPDATE).
2. La valeur NEXT n'est utilisable qu'en mode sortie avec des fichiers multivolumes sur bande. Pour traiter ces fichiers en mode entrée, il faut spécifier ANY ou une valeur numérique.

**ATTENTION**

Lorsque LABEL=NONE, FSN=1 est considéré comme ayant la valeur ANY ou NEXT.

**NBEFN**

Ce paramètre s'applique uniquement aux fichiers bande et disquette, en vue d'une concaténation. Sa valeur spécifie le nombre des fichiers à concaténer, le premier étant désigné par le paramètre FSN ou par "nom-fichier-externe".

Pour les fichiers bande (bandes multifichiers), la concaténation s'effectue dans l'ordre physique des fichiers sur le volume.

Pour les fichiers disquette, la concaténation s'effectue dans l'ordre des labels de fichiers en VTOC.

Dans le cas des fichiers multifragments, le processus est plus complexe : un fichier disquette ne pouvant comporter qu'un seul fragment par volume, multifragment implique multivolume. La concaténation commence par les fragments du fichier multifragment et enchaîne sur le fichier dont le label suit celui du dernier fragment traité. Il se peut donc que des fichiers se trouvant sur les volumes intermédiaires ne soient pas concaténés.

Lorsque NBEFN=ALL, tous les fichiers du volume sont concaténés. La concaténation s'arrête en fin de volume pour les bandes et en fin de VTOC pour les disquettes. Si plusieurs volumes sont spécifiés, l'opération continue jusqu'au dernier volume de la liste. Si MEDIA=\* est spécifié, l'opération s'arrête lorsque l'opérateur répond négativement au message demandant de monter le volume suivant.

**DENSITY**

Ce paramètre s'applique exclusivement aux fichiers bande ou cartouche ouverts en mode sortie. DENSITY permet de spécifier une densité d'enregistrement qui prévaut sur celle indiquée dans le label de fichier ; il est donc possible d'écrire sur la bande ou la cartouche en utilisant une densité d'enregistrement différente de celle du fichier existant. Pour une bande ou une cartouche multifichier, ce paramètre ne s'applique qu'à partir de l'écriture du premier fichier.

**Remarque :** En l'absence du paramètre DENSITY et quel que soit le mode de traitement utilisé, un message d'avertissement s'affiche lorsque la densité impliquée par le paramètre DEVCLASS diffère de celle indiquée dans le label de fichier, et c'est la densité inscrite dans le label qui est retenue.

**Exemples d'ordres ASSIGN***Exemple 1 :*

```
ASSIGN HJFILE, MY.FILE1;
```

Affectation du nom de fichier interne HJFILE au fichier catalogué MY.FILE1.

*Exemple 2 :*

```
ASSIGN DMFILE,Z.ABC, EXPDATE=30, DEVCLASS=MS/M500, MEDIA=B012;
```

Affectation du fichier disque non catalogué portant le nom de fichier externe Z.ABC. Si la réservation de place pour ce fichier (au moyen de l'ordre ALLOCATE) est effectuée dans l'activité en cours, la période de validité du fichier sera de 30 jours à partir de la date du jour.

***Exemple 3 :***

```
STEP S1,....;
ASSIGN SCRFIL, TMP1, TEMPRY, DEVCLASS=MT/T9/D1600,
      MEDIA=WORK, END=PASS;
.
.
.
ENDSTEP;
STEP S2,....;
ASSIGN INDATA, TMP1, TEMPRY;
.
.
.
ENDSTEP;
```

Un fichier de manoeuvre temporaire sur bande 9 pistes est conservé d'une activité à une autre. Le fichier est supprimé à la fin de l'activité S2. Noter qu'il est inutile de répéter les paramètres d'implantation (DEVCLASS et MEDIA) dans la seconde rubrique de description d'activité.

## 4.4 COMMENT

### 4.4.1 Fonction

Permet d'insérer un commentaire dans une description de travail.

### 4.4.2 Rubrique

Description de travail ou d'activité.

### 4.4.3 Format

```
COMMENT 'chaîne-commentaire' ;
```

### 4.4.4 Description

L'ordre COMMENT permet d'insérer un commentaire dans une description de travail. Des commentaires peuvent figurer à n'importe quel endroit d'une rubrique de description de travail, sauf dans une rubrique de description d'entrées. L'ordre COMMENT est un ordre fictif ; il n'est pas exécutable.

### 4.4.5 Paramètres

La chaîne commentaire est une chaîne protégée ; autrement dit, n'importe quel caractère peut y être utilisé et l'apostrophe doit être doublée lorsqu'elle ne délimite pas la chaîne. Dans le JOR, les apostrophes délimitant la chaîne font partie du commentaire.

#### *Exemples :*

```
COMMENT 'L''ACTIVITE SUIVANTE IMPRIME LES CHIFFRES DE VENTES  
MENSUELS' ;
```

```
COMMENT 'NOM ET ADRESSE DU PROPRIETAIRE' ;
```

## 4.5 CONSOLE

### 4.5.1 Fonction

Définit la destination des messages console de l'utilisateur.

### 4.5.2 Rubrique

Description de travail.

### 4.5.3 Format

```
CONSOLE nom-utilisateur;
```

### 4.5.4 Description

L'ordre CONSOLE s'applique à tous les ordres SEND qui suivent et à tous les messages opérateur émis par les modules chargeables utilisateur (au cours de l'exécution de l'activité) et par les procédures système (exécution d'ordres JCL étendus).

Cet ordre détermine également les consoles émettrice et destinatrice spécifiées dans les instructions COBOL "ACCEPT" et "DISPLAY". Pour plus de précisions sur ces instructions, voir le *Guide utilisateur COBOL*.

Les messages sont envoyés au terminal connecté au système sous l'identification d'utilisateur indiquée. Implicitement, le terminal destinataire est la console principale.

### 4.5.5 Paramètre facultatif

#### nom-utilisateur

Identifie l'utilisateur du terminal demandé. Le travail (dans le cas d'un ordre SEND) ou l'activité (dans les autres cas) est arrêté prématurément si l'identification de l'utilisateur n'est pas associée à un terminal connecté au système au moment où le message doit être envoyé. Si le nom d'utilisateur est omis, la console associée au travail en cours est implicite.

## 4.6 \$DATA

### 4.6.1 Fonction

Permet d'enregistrer des données d'entrée dans une unité de bibliothèque origine.

### 4.6.2 Rubrique

Hors d'une rubrique de description de travail.

### 4.6.3 Format

```

$DATA
{nom-unité-bibliothèque} [{(description-bibliothèque-de-sortie)}]
{[alphanum 17]*} {[SITE.IN}]

[ TYPE      = {DATA      } ]
[          {DATASSF} ]
[          {COBOL}  ]
[          {COBOLX} ]
[          {RPG}     ]
[          {FORTRAN} ]
[          {GPL}     ]
[          {JCL}     }

[FORMAT = (chiffres3,chiffres3,chiffres3,chiffres3)]

[END      = {EOF      } ]
[          {ENDDATA} ]
[          {'chaîne8'} ]

[ENDCHAR = 'chaîne1']

[CONTCHAR = 'chaîne1']

[RECSIZE  = { 110      } ]
[          {chiffres3} ]

[REPLACE]          [PRINT]          [CKSTAT]
[USER=nom-utilisateur] [PROJECT=nom-projet]

[BILLING=nom-compte];

```

#### 4.6.4 Description

Cet ordre permet de charger des données d'entrée dans une unité de bibliothèque origine. Il est traité directement par le lecteur des entrées et peut être considéré comme un travail indépendant comportant une seule activité, qui charge les données dans le sous-fichier de bibliothèque. Etant l'équivalent d'un travail, il ne peut figurer dans une rubrique de description de travail.

Le symbole \$ doit obligatoirement figurer en tête du nom de l'ordre.

Tous les articles qui suivent l'ordre \$DATA sont traités comme des articles de données à charger dans le fichier de sortie, jusqu'à ce qu'un indicateur fin soit détecté. Celui-ci peut être :

- un ordre \$ENDDATA,
- un article \$EOS,
- une marque fin de fichier (EOF),
- un article contenant la chaîne de caractères spécifiée dans END='chaîne8'.

La bibliothèque de sortie doit être un fichier permanent, accessible directement au moment de la lecture des entrées (c'est-à-dire sans attente de ressource ou de montage de volume).

Le nom d'utilisateur (USER) et le nom de projet (PROJECT) doivent satisfaire aux règles d'accès à la bibliothèque de sortie et aux règles de soumission des travaux. Un JOR est produit si le paramètre PRINT est spécifié ou si une erreur est détectée pendant le traitement des données d'entrée.

Si le nom d'unité de bibliothèque spécifié existe déjà dans la bibliothèque de sortie, les données n'y sont pas chargées, sauf si le paramètre REPLACE est également spécifié.

Le chargement de données dans un fichier peut également être effectué au moyen d'une rubrique standard de description de travail comprenant soit une activité CREATE (voir le *manuel sur les utilitaires de gestion de données*), soit une activité LIBMAINT (voir le *manuel de référence relatif à la maintenance des bibliothèques*). Noter cependant que par cette méthode, le transfert des données s'effectue pendant l'exécution des ordres JCL, alors qu'avec l'ordre \$DATA, il s'effectue pendant la lecture des entrées. L'ordre \$INPUT du lecteur des entrées charge les données d'entrée dans une unité de la bibliothèque SYS.IN, alors que l'ordre \$DATA permet de les charger directement dans une unité de fichier bibliothèque permanent.

Si un ordre \$JOB, \$INPUT ou \$SWINPUT figure dans une rubrique de description de données (c'est-à-dire entre les ordres \$DATA et \$ENDDATA) :

- \$INPUT est traité comme un article de données ;
- \$SWINPUT est exécuté si le paramètre CKSTAT est spécifié dans l'ordre \$DATA, ou traité comme un article de données s'il ne l'est pas ;
- \$JOB est traité comme un article de données si le train d'entrées est soumis à partir d'un fichier séquentiel, d'une unité de bibliothèque ou d'une station RBF ; dans les autres cas, il provoque un arrêt prématuré du traitement.

## 4.6.5 Paramètres

### 4.6.5.1 Paramètres obligatoires

#### **nom-unité-bibliothèque**

Ce premier paramètre positionnel, obligatoire, spécifie le nom de l'unité de bibliothèque dans laquelle les données d'entrée doivent être chargées. Il peut se présenter sous la forme d'une chaîne de 31 caractères maximum donnant le nom intégral de l'unité, ou sous la forme d'un nom de 17 caractères maximum suivi d'un astérisque (\*). Dans le dernier cas, le nom de l'unité est obtenu par enchaînement du nom précédant l'astérisque avec un groupe date-heure de 14 caractères généré par le système. Ce groupe date-heure est en format aammjj-hhmmss et précédé de la lettre D ; il représente les date et heure précises auxquelles le lecteur des entrées traite l'ordre \$DATA. Le groupe date-heure d'exécution permet de rendre uniques des noms d'unité qui seraient autrement identiques.

#### **description-bibliothèque-de sortie**

Ce second paramètre positionnel indique le nom de la bibliothèque dont est membre l'unité spécifiée par le premier paramètre positionnel. "description-bibliothèque-de-sortie" est facultatif et sa valeur implicite est SITE.IN (bibliothèque origine résidante). La bibliothèque spécifiée doit être permanente, et il est possible de spécifier tous les paramètres usuels d'une description de bibliothèque de sortie, à l'exception de SIZE.

### 4.6.5.2 Paramètres facultatifs

#### **USER**

Identifie l'utilisateur. Ce paramètre doit être spécifié lorsque la bibliothèque de sortie est cataloguée et protégée. Pour une description plus détaillée, voir l'ordre \$JOB.

#### **PROJECT**

Spécifie le nom du projet. Pour une description plus détaillée, voir l'ordre \$JOB.

#### **BILLING**

Spécifie le nom utilisé à des fins statistiques à l'intérieur du projet. Ce nom peut comporter jusqu'à 12 caractères (lettres, chiffres, trait d'union et caractère de soulignement) ; il doit commencer par une lettre ou un chiffre. La valeur implicite de ce paramètre est celle spécifiée pour le projet dans le catalogue de site. A défaut, le nom de projet est utilisé comme nom de compte dans la bannière du compte rendu d'exécution du travail.

## ATTENTION

Le paramètre BILLING doit être spécifié dans l'ordre \$JOB si aucune valeur implicite ne figure dans le catalogue de site.

### TYPE

Spécifie le format de l'unité de bibliothèque dans laquelle les données d'entrée doivent être chargées.

DATA indique que les articles chargés n'auront pas d'en-tête en format SSF et qu'il n'y aura pas d'articles de commande.

DATASSF indique que les articles chargés auront des en-têtes en format SSF et qu'il y aura des articles de commande.

Les autres valeurs concernent différents types de langages et sont décrites avec la commande MOVE dans le *Manuel de référence relatif à la maintenance des bibliothèques*.

### FORMAT

Définit la position des numéros de lignes et du texte pour un article. Les quatre valeurs spécifient respectivement le début du numéro de ligne, la fin du numéro de ligne, le début du texte et la fin du texte. Lorsque les deux premières valeurs sont égales à zéro, il n'y a pas de numéros de lignes. Lorsque les numéros de lignes sont spécifiés, ils doivent être cadrés à droite et les espaces sont interprétés comme autant de zéros.

FORMAT n'est pris en compte que si TYPE=DATASSF. S'il est omis, la valeur implicite est le format correspondant à la spécification de TYPE. Voir le *manuel de référence relatif à la maintenance des bibliothèques*.

### END

Spécifie le format de l'ordre marquant la fin des données d'entrée à charger.

La valeur implicite ENDDATA indique que les données d'entrée se terminent avec l'article précédent le premier ordre \$ENDDATA rencontré.

END='chaîne8' signifie que le dernier article d'entrée est celui qui se trouve immédiatement avant le premier article contenant uniquement cette chaîne de caractères, à partir de la première position.

END=EOF indique que le dernier article à charger est le dernier article du train d'entrées (c'est-à-dire l'article avant \$EOF).

L'article contenant \$ENDDATA, END='chaîne8' ou END=EOF n'est pas chargé dans la bibliothèque de sortie. Il doit être en code EBCDIC.

### ENDCHAR

Le caractère spécifié par ce paramètre est considéré comme délimiteur de fin d'article s'il constitue le dernier caractère significatif d'un article d'entrée. Ni ce caractère, ni les espaces de droite ne sont chargés dans l'article de sortie. Voir l'ordre \$INPUT pour des exemples d'utilisation de ce paramètre.

## CONTCHAR

Le caractère spécifié par ce paramètre est considéré comme marque de suite s'il constitue le dernier caractère significatif d'un article d'entrée ; l'article lu immédiatement après lui est donc enchaîné. Ni ce caractère, ni les espaces de droite ne sont chargés dans l'article de sortie. Voir \$INPUT pour des exemples d'utilisation de ce paramètre.

## RECSIZE

Spécifie la longueur d'article. Ce paramètre n'est pris en compte que lorsque les données d'entrée proviennent d'un terminal KDS. La valeur implicite est 110.

## REPLACE

Ce paramètre doit être spécifié lorsque les données à charger doivent se substituer au contenu d'une unité existante de la bibliothèque de sortie. Il doit en revanche être omis lorsque "nom-unité-bibliothèque" n'existe pas.

## PRINT

Demande la consignation des articles dans le JOR. Le maximum d'articles est de 500.

## CKSTAT

Ce paramètre doit être spécifié lorsque des ordres \$SWINPUT doivent figurer et être interprétés dans la rubrique de description de données.

### *Exemple 1 :*

```
$DATA SALES*, TOTAL, USER=ACCOUNT, PROJECT=TURNOVER  
      TYPE=DATA, PRINT;  
X X X X X X }  
. . . . . . }  
. . . . . . }  
. . . . . . }  
. . . . . . }  
X X X X X X }  
$ENDDATA;
```

Si la séquence ci-dessus est exécutée le 20 Juillet 1991 à 14 h 55 et 10 secondes, les données d'entrée sont chargées en tant qu'unité SALES910720-145510 de la bibliothèque permanente TOTAL. L'unité de bibliothèque n'existe pas avant l'exécution de cette séquence. Les articles d'entrée sont consignés dans le JOR.

**Exemple 2 :**

```

$DATA ABC, PROGLIB, USER=PROG, PROJECT=DEVELMT
  TYPE=JCL, FORMAT = (0, 0, 1, 80) REPLACE, PRINT;
X X X X X X }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
X X X X X X }
$ENDDATA;

```

données d'entrée

L'unité de bibliothèque ABC existait dans la bibliothèque permanente PROGLIB avant l'exécution de cette séquence. Le contenu de l'unité ABC est donc remplacé par les données fournies, qui consistent en des ordres JCL occupant les positions 1 à 80, et sans numéros de lignes. Les articles d'entrée sont consignés dans le JOR.

**Exemple 3 :**

```

$DATA ACCT, LIBRARY, USER=CHIEF, CKSTAT;
X X X X X X }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
X X X X X X }

```

données A

Contenu du fichier DATA-FILE

```

$SWINPUT DATA-FILE;           X X X X X X }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
X X X X X X }

```

données B

```

. . . . . }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
X X X X X X }

```

données C

Fin du fichier DATA-FILE

```

. . . . . }
. . . . . }
. . . . . }
. . . . . }
. . . . . }
$ENDDATA;

```

Les données doivent être chargées dans l'unité ACCT, qui est un nouveau sous-fichier de la bibliothèque LIBRARY. Le paramètre CKSTAT prévient le lecteur des entrées que des ordres \$SWINPUT figurent dans la rubrique de description de données et qu'il doit les exécuter au fur et à mesure.

Les données sont chargées dans l'ordre suivant :

données A  
données B  
données C

## 4.7 DEFINE

### 4.7.1 Fonction

Demande la journalisation pour protéger les fichiers, spécifie les points de reprise dans les fichiers et les paramètres de traitement des fichiers.

Les paramètres spécifiés dans l'ordre DEFINE complètent ceux figurant dans la description de fichier du programme utilisateur et/ou dans certains ordres JCL, et prévalent sur eux. Ils permettent également de compléter la description de fichier dans un label de fichier.

### 4.7.2 Rubrique

Description d'activité.

### 4.7.3 Format

```
DEFINE      nom-fichier-interne
            [ JOURNAL={AFTER  } ]
            [             {BEFORE} ]
            [             {NONE   } ]
            [             {BOTH   } ]

            [ CKPTLIM={chiffres8  [EOV]  } ]
            [             {NO          } ]
            [             {EOV         } ]

            [ NBBUF=chiffres3 ]
            [ BPB=chiffres3 ]
            [ {SYSOUT} ]
            [ {NSYSOUT} ]

            [ ERROPT={SKIP      } ]
            [             {ABORT    } ]
            [             {IGNORE   } ]
            [             {RETCODE  } ]

            [ WRCHECK ]
            [ OPTIMIZE ]
            [ {BPIOC} ]
            [ {FRIOC} ]
            [ {RAHEAD} ]
```

```

[KEYLOC=chiffres5      ]
[INKEYLOCK=chiffres5]

[ {DLREC } ]
[ {NDLREC} ]
[ {COMPACT } ]
[ {NCOMPACT} ]
[ {TRUNCSSF} ]
[ {NTRNCSSF} ]
[FPARAM]

[LTRKSIZE = chiffres5]

[FILEORG={SEQ      }]
[      {RELATIVE} ]
[      {INDEXED} ]
[      {LINKQD} ]
[      {NONE     } ]

[KEYSIZE=chiffres3]

[CISIZE=chiffres5]

[CIFSP=chiffres2]

[BUFPOOL=alphanum4]

[READLOCK={NORMAL } ]
[      {EXCL   } ]
[      {STAT   } ]

[LOCKMARK]

[FILEFORM={UFAS } ]
[      {ANSI } ]
[      {NSTD } ]
[      {LINKQD} ]
[      {NONE } ]

[BLKSIZE=chiffres5]

[RECSIZE=chiffres5]

[ {RECFORM={F   } }
[ {      {V   } } ]
[ {      {U   } } ]
[ {      {FB  } } ]
[ {      {VB  } } ]

[FUNCTIONMASK=hexa_8]

[BLKOFF=chiffres3]

[ {CONVERT } ]
[ {NCONVERT} ]

[DATACODE={BCD     }]
[      {ASCII  } ]
[      {EBCDIC} ]

[COLLATE={BCD     }]
[      {ASCII  } ]

```

```

[          {EBCDIC} ]
[ {BSN  } ]
[ {NBSN} ]
[NSORTIDX]
[ {PRINTER =(options-imprimante)} ]
[ {TN =(options-terminal  )} ];

```

**"options-imprimante"** a le format suivant :

```

[ {SLEW  } ]
[ {NSLEW} ]
[ MARGIN=chiffres3]
[ PRDEN={6} ]
[      {8} ]
[ FORMHT=chiffres3]
[ HOF ={chiffres3} ]
[      {1      } ]
[ FF1={chiffres3      } ]
[      {valeur-FORMHT} ]
[ FF2={chiffres3} ]
[      {0      } ]
[ CHi=[chiffres3      [chiffres3] ...]

```

**"options-terminal"** a le format suivant :

```

[ {PROMPT=car12} ]
[ {NPROMPT} ]
[ EOF=car4 ]
[ {MSG} ]
[ {NMSG} ]
[ {SLEW} ]
[ {NSLEW} ]
[ MARGIN=chiffres3]
[ PRDEN={6} ]
[      {8} ]
[ FORMHT=chiffres3]

```

```

[HOF={chiffres3}]
[ 1      ]
[FF1={chiffres3}]
[ valeur-FORMHT ]
[FF2={chiffres3}]
[ 0      ]
[CH1=(chiffres3  [chiffres3] ...)]

```

#### 4.7.4 Description

L'ordre DEFINE permet de demander la journalisation, de créer des points de reprise dans les fichiers et de spécifier des paramètres d'optimisation du traitement des fichiers (par exemple, le nombre des tampons).

Les paramètres spécifiés dans l'ordre DEFINE complètent ceux figurant dans la description de fichier du programme utilisateur et/ou dans certains ordres JCL (en particulier, les ordres relatifs au traitement des E/S) et prévalent sur eux. Les paramètres de l'ordre DEFINE n'ont pas de valeur implicite ; si l'un d'entre eux n'est pas spécifié, c'est donc la valeur correspondante éventuellement indiquée dans la description du fichier ou dans le JCL qui s'applique.

A chaque ordre DEFINE (sauf ceux associés à des fichiers SYSOUT standard) doit correspondre un ordre ASSIGN le précédent dans la même rubrique de description d'activité.

Un ordre DEFINE associé au premier fichier d'une série de fichiers concaténés s'applique également aux fichiers suivants de la série.

Un ordre DEFINE peut toutefois ne porter que sur le premier fichier d'une série, lorsque les autres n'ont pas de noms de fichiers internes uniques au moyen desquels l'ordre DEFINE pourrait y faire référence.

Les valeurs de "dérogation" s'appliquent au fichier portant le nom interne spécifié à l'ouverture du fichier, et restent effectives jusqu'à la libération du fichier. Pour les fichiers disque, ou pour les fichiers bande ouverts en entrée ou en adjonction, les paramètres de l'ordre DEFINE ne l'emportent sur les spécifications du label de fichier que si FILEFORM=NSTD.

Pour les fichiers bande non catalogués traités en mode sortie les informations de niveau fichier ne prévalent pas sur les valeurs de paramètres de l'ordre DEFINE.

Un ordre DEFINE associé à un sous-fichier SYSIN standard n'est pas pris en compte.

De façon générale, aucune valeur spécifiée n'est contrôlée par rapport à la méthode d'accès utilisée dans le programme. Il peut donc arriver que certains paramètres indiqués soient inutiles et par conséquent non pris en compte. Par exemple :

- les valeurs indiquées dans un ordre DEFINE et celles fournies dans le programme COBOL correspondant ne font pas l'objet d'un contrôle de cohérence ;
- le paramètre NBBUF de DEFINE peut spécifier moins ou davantage de tampons qu'il n'est indiqué dans la définition de fichier du programme. Si NBBUF en spécifie davantage, un arrêt abnormal de l'activité peut se produire.

## 4.7.5 Paramètres

### 4.7.5.1 Paramètre obligatoire

#### **nom-fichier-interne**

Ce paramètre indique le fichier auquel s'applique l'ordre DEFINE. Il comporte un maximum de huit caractères (lettres, chiffres, trait d'union et caractère de soulignement).

### 4.7.5.2 Paramètres facultatifs

#### **JOURNAL**

Ce paramètre provoque l'enregistrement de l'image de blocs de données dans un fichier journal ou plus. Les fichiers journaux sont utilisés avec la relance sur point de reprise (voir chapitre 1). L'image du journal sert à restaurer les blocs de données dans leur état antérieur à l'arrêt anormal.

Si le fichier est traité en mode entrée (ACCESS=READ, SPREAD ou ALLREAD), les journaux ne sont pas utilisés.

**AFTER** Les images des blocs de données sont celles du journal Après (articles utilisateur après modification). Valable uniquement pour les fichiers UFAS et IDS/II.

**BEFORE** Les images des blocs de données sont celles du journal Avant (images des articles utilisateur avant modification, enregistrées depuis le dernier point de reprise rencontré). Valable pour les fichiers UFAS et IDS/II.

**NONE** Aucun fichier journal n'est généré.

**BOTH** Les journaux Avant et Après sont tous deux générés. Valable uniquement pour les fichiers UFAS et IDS/II.

Le type de journal à utiliser lors du traitement d'un fichier peut être spécifié dans le catalogue. Si c'est le cas et qu'une option JOURNAL est en outre spécifiée dans l'ordre DEFINE, les règles de dérogation suivantes s'appliquent :

**Règle 1 :** Cette règle vaut si les droits d'accès n'ont pas été mis en oeuvre, ou bien s'ils ont été mis en oeuvre et que l'utilisateur dispose au moins du droit d'accès RECOVERY ou OWNER pour le fichier concerné.

C'est la valeur donnée dans DEFINE qui prévaut sur celle du catalogue.

**Règle 2 :**

Cette règle vaut si les droits d'accès ont été mis en oeuvre et que l'utilisateur ne dispose pas du droit RECOVERY ou OWNER pour le fichier concerné (son droit d'accès est alors LIST, EXECUTE, READ ou WRITE).

La valeur donnée dans DEFINE n'est prise en compte que si elle spécifie le même niveau de protection ou un niveau plus élevé que celui spécifié dans le catalogue, c'est-à-dire en résumé dans les cas suivants :

Valeur spécifiée dans DEFINE	Résultat
Valeur identique à celle du catalogue	OK
BOTH	OK
Valeur quelconque ou valeur du catalogue = No	OK
Tous les autres cas	Erreur dans ASSIGN ou DEFINE suivant la séquence DEFINE --> ASSIGN ou ASSIGN --> DEFINE

**CKPTLIM**

Ce paramètre indique quand doivent être automatiquement constitués des points de reprise. Ce mécanisme est indépendant du programme COBOL (qui comporte des appels à la fonction points de reprise). Ce paramètre peut prendre l'une des valeurs suivantes :

NO	Pas de constitution de points de reprise.
EOV	Dans le cas d'un fichier disque ou bande multivolume pour lequel tous les volumes n'ont pas été montés initialement, demande la constitution d'un point de reprise à la fin de chaque volume.
chiffres8	Cette valeur spécifie la fréquence de constitution des points de reprise, un point de reprise étant constitué tous les "chiffres8" articles. Cette valeur peut être spécifiée en même temps qu'EOV.

Si des fichiers sous GAC sont utilisés par l'activité COBOL, une consolidation automatique est exécutée à la place du point de reprise. Pour plus d'informations sur les points de reprise, voir le paragraphe 3.3.6 du *Guide utilisateur de GAC-Extended*.

**NBBUF**

Ce paramètre indique le nombre de tampons devant être affectés au fichier.

**BPB**

Ce paramètre sert à définir le nombre de blocs par tampon (255).

**SYSOUT**

S'il est spécifié, ce paramètre instaure l'utilisation du mécanisme SYSOUT pour l'écriture d'un fichier, à condition que les caractéristiques de celui-ci (RECSIZE) soient acceptables (voir l'ordre SYSOUT). Ce paramètre peut par exemple être utilisé lorsque la clause ASSIGN...TO...-SYSOUT ne figure pas dans le programme COBOL.

**Remarque :** L'utilisation du paramètre SYSOUT dans l'ordre DEFINE équivaut à la spécification de -SYSOUT dans un programme COBOL.

**NSYSOUT**

S'il est spécifié, ce paramètre annule toute demande d'utilisation du mécanisme SYSOUT (notamment dans le programme ou par l'ordre SYSOUT), qui ne sera donc pas employé pour l'écriture du fichier.

Ce paramètre doit être spécifié lorsque l'utilisateur veut empêcher le système d'utiliser le mécanisme SYSOUT pour la création d'un fichier SYSOUT permanent non catalogué sur bande (l'utilisation du mécanisme SYSOUT dans ce cas entraînerait une longueur d'article supérieure à 600 octets), sauf si TAPE=NSYSOUT a été spécifié dans la directive de configuration OWDFLT.

**ERROPT**

Ce paramètre indique au système la mesure à prendre en cas d'erreur d'E/S.

SKIP	Cette valeur s'applique uniquement en mode séquentiel, et signifie que l'article n'est pas pris en compte.
ABORT	L'activité est arrêtée prématulement (code de gravité 3).
IGNORE	Cette valeur n'est utilisée que pour les bandes. L'erreur d'E/S n'est pas prise en compte et les données renvoyées consistent en la partie lue de l'article, complétée à droite par des caractères conventionnels. Pour un accès de type séquentiel ou dynamique, le pointeur d'article courant désigne l'article valide suivant.
RETCODE	Le système génère un code signalant une erreur d'E/S (par exemple IOFAIL) et le traitement du fichier n'est pas poursuivi. Le pointeur d'article courant est non défini dans le cas d'un accès de type séquentiel ou dynamique.

**WRCHECK**

Lorsque ce paramètre est spécifié pour l'activité en cours, chaque bloc écrit dans le fichier est contrôlé au moment de l'écriture (pour les fichiers cloisonnés uniquement).

**OPTIMIZE**

Permet un accès rapide en lecture et en écriture pour les fichiers UFAS séquentiels, . Pour plus d'informations, voir le *Guide utilisateur de l'optimiseur de traitement par lots UFAS*.

### **BPIOC**

Si BPIOC est spécifié, la mise en antémémoire des fichiers est ignorée. Ce paramètre prévaut sur toutes les autres options d'antémémoire indiquées dans le catalogue ou dans les paramètres d'affectation. Pour plus d'informations sur la mise en antémémoire des fichiers, voir le *Guide utilisateur relatif à l'antémémoire LMC*.

### **FRIOC**

Si FRIOC est spécifié, les fichiers sont chargés dans l'antémémoire. Ce paramètre prévaut sur toutes les autres options d'antémémoire indiquées dans le catalogue ou dans les paramètres d'affectation. Pour plus d'informations sur la mise en antémémoire des fichiers, voir le *Guide utilisateur relatif à l'antémémoire LMC*.

### **RAHEAD**

Par défaut, pour les fichiers séquentiels chargés en antémémoire, la technique de "lecture anticipée" est utilisée lorsqu'un bloc de fichiers est référencé pour la première fois, afin que les 16 blocs suivants soient également lus. RAHEAD permet de forcer l'utilisation de cette technique, que le fichier soit séquentiel ou non. Pour plus d'informations sur la mise en antémémoire des fichiers, voir le *Guide utilisateur relatif à l'antémémoire LMC*.

### **KEYLOC**

Ce paramètre s'applique aux fichiers UFAS indexés. Il définit la position relative (à partir de la gauche) du premier caractère de la clé d'article des données dans l'article. Le premier octet de l'article est 0.

**Remarque :** Il existe une différence d'un octet entre le paramètre KEYLOC décrit dans l'ordre JCL de base DEFINE ou dans le groupe de paramètres OUTDEF, et le paramètre KEYLOC défini dans l'ordre PREALLOC. Le premier octet d'un article est 0 pour DEFINE et OUTDEF, et 1 pour PREALLOC.

### **DLREC**

Les articles supprimés (ceux commençant par "FF"X sont à supprimer) sont autorisés dans les fichiers cloisonnés.

### **NDLREC**

Les articles supprimés ne sont pas autorisés dans les fichiers cloisonnés.

### **NTRNCSSF**

En format SSF, l'en-tête et les articles de commande SSF ne sont pas supprimés.

## **COMPACT**

Les articles doivent être enregistrés sur le support sous leur forme compacte (avec tassement des espaces).

## **NCOMPACT**

Les articles doivent être enregistrés sur le support sans tassement des espaces.

## **FPARAM**

Ce paramètre spécifie la prévalence des valeurs spécifiées dans l'ordre DEFINE sur certaines valeurs spécifiées par ailleurs.

## **LTRKSIZE**

Définit la taille de piste logique pour le fichier.

## **FILEORG**

Définit l'organisation du fichier, correspondant à l'une des valeurs suivantes :

<b>Abréviation</b>	<b>Organisation du fichier</b>
SEQ	séquentiel
RELATIVE	relatif
INDEXED	indexé
LINKQD	cloisonné chaîné
NONE	organisation indéterminée

## **KEYSIZE**

Ce paramètre donne la longueur en octets de la clé d'article. La valeur maximum autorisée est 255.

## **CISIZE**

Ce paramètre donne, pour les fichiers UFAS, la longueur en octets de l'intervalle de contrôle (CI).

## **CIFSP**

Ce paramètre indique, pour les fichiers UFAS indexés, la place libre dans l'intervalle de contrôle (en pourcentage).

**BUFPOLL**

Ce paramètre identifie le pool de tampons (groupe de tampons de même nom) utilisé pour un fichier UFAS sur disque.

**READLOCK**

Ce paramètre s'applique uniquement si SHARE=MONITOR a été spécifié pour un fichier UFAS ou IDS/II sur disque. Il définit un type de verrouillage au niveau du CI (et non au niveau du fichier). Les verrous définis par READLOCK sont levés à la fin de l'unité de consolidation dans laquelle ils apparaissent. Pour plus d'informations, voir le *Guide utilisateur de GAC-Extended*.

READLOCK peut prendre l'une des valeurs suivantes :

NORMAL	Permet les accès simultanés suivants :
	(i) (a) plusieurs accès NORMAL en lecture ou (b) un seul accès NORMAL en écriture ET (ii) plusieurs accès STAT en lecture ET (iii) aucun accès EXCL (tant en lecture qu'en écriture)
EXCL	Permet les accès simultanés suivants :
	(i) (a) un seul accès EXCL en lecture ou (b) un seul accès EXCL en écriture ET (ii) aucun accès NORMAL (ni en lecture, ni en écriture) ET (iii) plusieurs accès STAT en lecture
STAT	Permet les accès simultanés suivants :
	(i) (a) plusieurs accès NORMAL en lecture ou (b) un seul accès NORMAL en écriture ou (ii) (a) un seul accès EXCL en lecture ou (b) un seul accès EXCL en écriture ET (iii) plusieurs accès STAT en lecture

En utilisant la convention booléenne, ces règles peuvent être résumées comme suit :

NORMAL	= (ia OU ib) ET (ii) ET (iii)
EXCL	= (ia OU ib) ET (ii) ET (iii)
STAT	= (ia OU ib) OU ((ia OU ib) ET (iii))

NORMAL et EXCL s'excluent mutuellement pour un CI donné, à un moment donné. Les accès STAT en lecture peuvent toujours avoir lieu sans être affectés par les mises à jour simultanées.

### **LOCKMARK**

Ce paramètre s'applique uniquement si SHARE=MONITOR a été spécifié pour un fichier UFAS sur disque. Il permet de conserver le pointeur d'article courant entre deux unités de consolidation successives.

Pour plus de précisions sur ce paramètre, voir le *Guide utilisateur de GAC-Extended*.

### **FILEFORM**

Ce paramètre indique le format du fichier :

UFAS	Fichier UFAS.
LINKQD	Fichier cloisonné chaîné.
ANSI	Fichier bande en format ANS, dont les données sont écrites en code ASCII.
NSTD	Fichier bande de format inconnu. Les données sont délivrées par blocs, marques de bande comprises. Les labels ne sont pas traités par les routines OPEN et CLOSE.
NONE	Le format n'est pas défini.

### **BLKSIZE**

Spécifie la longueur de bloc en octets.

### **RECSIZE**

Spécifie la longueur d'article en octets.

### **RECFORM**

Spécifie le format d'article, correspondant à l'une des valeurs suivantes :

<b>Abréviation</b>	<b>Format d'article</b>
F	articles de longueur fixe, non groupés en blocs
V	articles de longueur variable, non groupés en blocs
U	articles de format indéterminé
FB	articles de longueur fixe, groupés en blocs
VB	articles de longueur variable, groupés en blocs

### **FUNCMASK**

Ce paramètre sert à sélectionner les fonctions du contrôleur de bande (par exemple, option de groupage/dégroupage) pour les fichiers UFAS sur bande. Le paramètre FUNCMASK est en particulier appliqué aux fichiers bande BFAS non standard pour lesquels le message "MEDIA RECOGNITION IMPOSSIBLE" (reconnaissance du support impossible) a été généré par GCOS. Le masque indiqué, qui contient huit caractères hexadécimaux, est chargé dans le contrôleur de bande au moment de l'ouverture du fichier. Pour plus de précisions sur ce paramètre, voir le *Guide utilisateur d'UFAS-EXTENDED*.

### **BLKOFF**

Ce paramètre permet au programme d'ignorer les informations non désirées qui figurent au début de chaque bloc dans certains fichiers bande (notamment les fichiers bande ANS - version 1). La valeur indiquée définit la longueur des données non souhaitées (déplacement par rapport au début du bloc) et correspond à la position relative dans chaque bloc du premier octet des données voulues.

**Remarque :** La valeur de BLKSIZE doit inclure le déplacement éventuel par rapport au début du bloc.

### **CONVERT**

Demande la conversion du code de données pour les fichiers bande.

### **NCONVERT**

Demande que le code de données ne soit pas converti pour les fichiers bande.

### **DATACODE**

Ce paramètre définit le code de données. Les valeurs possibles sont BCD, ASCII et EBCDIC.

### **COLLATE**

Ce paramètre spécifie l'ordre de classement pour les fichiers UFAS sur disque. Les valeurs possibles sont BCD, ASCII et EBCDIC.

### **BSN**

Indique que les blocs d'un fichier bande doivent être numérotés.

### **NBSN**

Indique que les blocs d'un fichier bande ne doivent pas être numérotés.

## **NSORTIDX**

Désactive la mise à jour des index secondaires des fichiers UFAS indéxés dans le mode de traitement UPDATE. Dans ce cas, la fonction SORT\_INDEX doit être exécutée après la fermeture du fichier afin de le restaurer en toute sécurité, car le statut du fichier est instable.

**PRINTER=options-imprimante**  
**TN=options-terminal**

**PRINTER=options-imprimante**

Ces options permettent de spécifier les paramètres d'impression des fichiers.

## **SLEW**

Exécution des instructions de saut de papier à l'impression du fichier.

## **NSLEW**

Lorsque le fichier est imprimé par l'éditeur des sorties, toute fonction de saut de papier est transformée en passage à la ligne suivante.

## **MARGIN**

Indique le nombre d'espaces à insérer au début de chaque ligne d'impression. La valeur implicite est zéro (0). La valeur spécifiée est limitée par la longueur maximum de la ligne d'impression sur l'imprimante utilisée.

## **PRDEN**

Indique la densité verticale d'impression, qui peut être de six ou huit lignes par pouce. La valeur implicite est celle autorisée par le type de papier spécifié dans MEDIA ; si MEDIA n'est pas spécifié, la valeur implicite est celle de l'imprimante implicite de l'installation.

La valeur du paramètre FORMHT est exprimée en nombre de lignes par page. Si une page est définie comme devant contenir n lignes, sa hauteur sera donc de n/6 si PRDEN=6 ou de n/8 si PRDEN=8.

Si PRDEN est utilisé pour optimiser l'impression, le paramètre FORMHT doit être modifié en conséquence. Ainsi, si PRDEN=6 et FORMHT=54, la page fera 9 pouces. Si PRDEN doit prendre la valeur 8, FORMHT doit prendre la valeur 72 afin que les 9 pouces continuent à être entièrement utilisés ; sinon, les 54 lignes n'occuperaient que 7 pouces environ.

## **FORMHT**

Indique la hauteur de feuillet en nombre de lignes. La valeur maximum est 255. Voir le paramètre PRDEN ci-dessus.

## **HOF**

Indique le numéro de la première ligne à imprimer sur le feuillet. La valeur maximum est 255, mais voir également la remarque ci-après.

## **FF1**

Premier seuil page pleine : indique le numéro de la dernière ligne à imprimer sur le feuillet. La valeur maximum est 255, mais voir également la remarque ci-après.

## **FF2**

Second seuil page pleine : un code retour est émis lorsque cette ligne est atteinte ou dépassée. La valeur maximum est 255, mais voir également la remarque ci-après.

**Remarque :** Les restrictions suivantes sont applicables aux options d'édition :

1 <= HOF <= FF2 <= FFFF1 <= FORMHT

## **CHi**

La bande pilote d'une imprimante est une bande physique comportant des canaux, qui sont perforés dans certaines positions pour commander les mouvements verticaux du papier.

Les imprimantes du DPS 7000 ne comportant pas de bande pilote physique, l'avancement du papier est commandé par une bande pilote simulée. Le résultat est le même qu'avec une bande pilote standard de 12 canaux, dans la limite de 20 sauts par feuillet, répartis sur les 12 canaux. L'utilisateur peut demander un ou plusieurs sauts, en ordre croissant, pour chaque paramètre CHi ( $1 < i < 12$ ). Chaque saut indique un numéro de ligne (jusqu'à trois chiffres) dont la valeur doit être inférieure ou égale à la valeur de FORMHT. Le nombre maximum de sauts pour l'ensemble des paramètres CHi est de 20.

### ***Exemple :***

CH1=(10, 25, 50), CH2=(5, 15)

Ce paramètre indique des sauts aux lignes 10, 25 et 50 sur le canal 1 et aux lignes 5 et 15 sur le canal 2.

## **TN=options-terminal**

Les options suivantes permettent la spécification de paramètres pour les fichiers sur terminal. Pour plus de détails, voir le *Guide utilisateur IOF*.

## **PROMPT**

Définit une chaîne de 12 caractères maximum qui s'affichera comme message de guidage.

**NPROMPT**

Indique qu'aucun message de guidage ne sera utilisé.

**EOF**

Définit une chaîne de quatre caractères utilisés comme indicateur de fin de fichier en entrée.

**NMSG**

Tous les messages envoyés à la console dont l'émetteur est autre que l'activité interactive sont placés en attente jusqu'à la fermeture du fichier.

**MSG**

Les messages envoyés à la console ne sont pas mis en attente.

Les options-terminal restantes ont la même signification que les options-imprimantes du même nom.

## 4.8 \$ENDDATA

### 4.8.1 Fonction

Indique la fin d'une rubrique de description de données.

### 4.8.2 Rubrique

Hors d'un travail.

### 4.8.3 Format

\$ENDDATA ;

### 4.8.4 Description

Cet ordre clôture une rubrique de description de données commençant par un ordre \$DATA, lorsque le paramètre END est omis dans l'ordre \$DATA. L'article contenant \$ENDDATA n'est pas considéré comme un article de données à charger par \$DATA. Le symbole \$ est obligatoire et doit figurer à la première position de l'article. L'ordre \$ENDDATA dans son ensemble, point-virgule compris, doit former un seul article.

## 4.9 \$ENDINPUT

### 4.9.1 Fonction

Indique la fin d'une rubrique de description d'entrées.

### 4.9.2 Rubrique

Description de travail.

### 4.9.3 Format

```
$ENDINPUT [nom-rubrique-de-description-d'entrées] ;
```

### 4.9.4 Description

L'ordre \$ENDINPUT marque la fin d'une rubrique de description d'entrées (voir l'ordre \$INPUT). Lorsque le lecteur des entrées reconnaît cet ordre, le sous-fichier SYSIN contenant les articles de la rubrique de description d'entrées est fermé.

Le symbole \$ est obligatoire à la première position de l'article.

### 4.9.5 Paramètre facultatif

#### nom-rubrique-de-description-d'entrées

Ce paramètre n'a de signification que si END=MATCH a été déclaré dans l'ordre \$INPUT correspondant à cet ordre \$ENDINPUT. Si les deux noms de rubriques de description d'entrées coïncident, \$ENDINPUT est considéré comme l'ordre de fermeture de la rubrique de description d'entrées.

L'ordre \$ENDINPUT est traité en tant que donnée de la rubrique de description d'entrées en cours si "nom-rubrique-de-description-d'entrées" n'est pas identique à celui indiqué dans un ordre \$INPUT antérieur.

## 4.10 \$ENDJOB

### 4.10.1 Fonction

Indique la fin d'une rubrique de description de travail.

### 4.10.2 Rubrique

Description de travail.

### 4.10.3 Format

\$ENDJOB ;

### 4.10.4 Description

L'ordre \$ENDJOB est obligatoirement le dernier ordre d'un travail commençant par un ordre \$JOB.

L'ordre \$JOB n'est pas obligatoire pour les travaux soumis au moyen de l'ordre RUN ou de la commande opérateur EJR, et lorsque l'ordre \$JOB est omis, l'ordre \$ENDJOB peut également l'être.

Lorsqu'un travail est soumis au système, le lecteur des entrées reconnaît son ordre \$ENDJOB et ferme le fichier contenant les ordres JCL origine (voir l'ordre \$JOB).

Si l'ordre \$ENDJOB est omis à tort, la traduction du JCL est effectuée, mais le travail s'arrête prématurément. Un JOR est produit une fois la traduction du JCL terminée.

Le symbole \$ est obligatoire à la première position de l'article.

L'ensemble de l'ordre \$ENDJOB doit former un seul article.

## 4.11 ENDSTEP

### 4.11.1 Fonction

Demande l'exécution du module chargeable mentionné dans l'ordre STEP correspondant. C'est le dernier ordre d'une rubrique de description d'activité.

### 4.11.2 Rubrique

Description d'activité.

### 4.11.3 Format

ENDSTEP;

### 4.11.4 Description

L'ordre ENDSTEP est le dernier ordre d'une rubrique de description d'activité (voir l'ordre STEP pour la présentation de ces rubriques). Il est obligatoire.

Dans le cadre de l'exécution d'un travail, l'ordre ENDSTEP marque la fin des procédures de lancement d'une activité (affectation de ressources) et demande l'exécution du module chargeable (programme) désigné dans l'ordre STEP correspondant.

## 4.12 EXDIR

### 4.12.1 Fonction

Soumet pour exécution les commandes GCL réservées à l'opérateur principal dans le cadre d'un travail en traitement par lots.

### 4.12.2 Rubrique

Description de travail.

### 4.12.3 Format

```
EXDIR 'commande[ ;commande]...';
```

### 4.12.4 Description

Les commandes opérateur principal (MAIN) décrites dans le *Guide de l'opérateur système* peuvent être soumises au moyen d'un ordre EXDIR si le travail de traitement par lots dans lequel elles apparaissent est rattaché à un projet doté de l'attribut MAIN. Si tel n'est pas le cas, seules les directives IOF présentées dans le *manuel de référence IOF* peuvent être soumises.

La longueur maximum de la chaîne de commandes est 255 octets. Les commandes qui la composent doivent être séparées par un point-virgule.

L'ordre EXDIR ne peut pas soumettre les directives GCL MWINLIB BIN, BYE et EXDIR.

L'image des commandes soumises ainsi que les éventuels messages d'erreur figurent dans le JOR.

## 4.13 EXECUTE

### 4.13.1 Fonction

Traduit et exécute dans le cadre d'une activité une séquence d'ordres JCL enregistrés.

#### 4.13.2 Rubrique

Description de travail.

#### 4.13.3 Format

```
EXECUTE { [INFILE=] (description-fichier-d'entrée-séquentiel)
          {nom-unité-bibliothèque [INLIB=] (description-
          bibliothèque-d'entrée)}
          [VALUES=( [valeur-paramètre1] [valeur-paramètre2]... )]
          [      [mot-clé1=valeur-mot-clé1]      ]
          [      [mot-clé2=valeur-mot-clé2]...      ]
          [PRTFILE=(description-fichier-impression) ]
          [      [PRTDEF=(paramètres-définition)]      ]
          [PRTOUT=(paramètres-sysout) ]
          [SIZEOPT=(paramètres-size) ]
          [STEPOPT=(paramètres-step) ];
```

#### 4.13.4 Description

L'ordre EXECUTE crée une activité qui utilise le traducteur JCL pendant l'exécution d'un travail pour traduire une séquence enregistrée d'ordres JCL. Une fois traduits, ces ordres sont soumis pour exécution. Lorsqu'ils sont exécutés, le traitement du travail de base reprend à l'ordre qui suit EXECUTE. En résumé, EXECUTE appelle dynamiquement une série d'ordres JCL au cours de l'exécution d'un travail.

Une séquence JCL traitée par EXECUTE peut contenir d'autres ordres EXECUTE. Chacun d'entre eux traduit puis exécute des ordres JCL.

Une séquence JCL peut également faire référence à des rubriques de description d'entrées figurant dans la description du travail de base, et recevoir des valeurs de paramètres de l'ordre EXECUTE (paramètre VALUES).

Un compte rendu de traduction du JCL est généré pour chaque ordre EXECUTE. Implicitement, ce compte-rendu est imprimé comme un fichier SYSOUT standard, mais l'utilisateur peut demander qu'il soit enregistré dans le fichier spécifié.

**Remarque :** L'ordre EXECUTE définit une activité entière, mais il est considéré comme faisant partie du système d'exploitation de base et en tant que tel, est traité dans le présent manuel comme un ordre JCL de base.

### Commentaires et restrictions

Il importe de bien faire la distinction entre **INVOKE**, ordre interprété au moment de la traduction du JCL, et **EXECUTE**, ordre lancé au moment de l'exécution.

**INVOKE** est statique en ce sens qu'une séquence d'ordres JCL est insérée dans une description de travail au moment de la traduction, et en devient ainsi partie intégrante.

**EXECUTE** est au contraire dynamique car la séquence à exécuter est identifiée et traduite seulement au moment de l'exécution, et une fois exécutée, n'a plus aucune signification pour la description de travail contenant l'ordre **EXECUTE**.

Les trois premiers points énumérés ci-après découlent de cette différence entre l'ordre **INVOKE** et l'ordre **EXECUTE**.

- Si un ordre **EXECUTE** donné est exécuté plusieurs fois dans un travail (par exemple suite à un ordre **JUMP**), il est possible de créer chaque fois une version différente de la séquence d'ordres JCL (si, par exemple, le fichier contenant la séquence doit être modifié avant chaque exécution d'**EXECUTE**).
- Lorsqu'une liste de bibliothèques à explorer est définie dans le travail contenant l'ordre **EXECUTE**, elle n'est pas disponible lors de l'exécution des ordres JCL enregistrés. S'il faut explorer des bibliothèques (par exemple, pour exécuter un ordre **LIBMAINT** contenu dans la séquence d'ordres JCL), la liste doit en être déclarée dans la séquence même (au moyen de l'ordre de gestion de bibliothèques "LIB").
- Aucun ordre **PREFIX** figurant dans le travail de base n'est valable pour la séquence JCL enregistrée. Si l'identification du projet auquel est rattaché le travail ne doit pas servir de préfixe pour la séquence, il faut spécifier un ordre **PREFIX** dans la séquence même.
- Le fichier séquentiel contenant la séquence à traduire et exécuter doit être en format **DATA** ou **DATASSF**.
- Les ordres reconnus par le lecteur des entrées (**\$JOB**, **\$ENDJOB**, **\$INPUT**, **\$ENDINPUT**, **\$DATA**, **\$ENDDATA** et **\$SWINPUT**) ne doivent pas figurer à l'intérieur d'une séquence JCL appelée par **EXECUTE**. La séquence ne doit donc pas contenir de rubrique de description d'entrées.
- La séquence doit contenir au moins une rubrique de description d'activité complète. Noter que tous les ordres JCL étendus (à l'exception de **SORTWORK**), ainsi que les ordres JCL de base **ATTACH**, **EXECUTE** et **RUN** peuvent être considérés comme des rubriques de description d'activité complètes.

- Si une relance à chaud a lieu pendant l'exécution de l'ordre EXECUTE, elle s'effectue toujours depuis le début de l'activité (la traduction est donc entièrement refaite). Si une relance à chaud a lieu pendant l'exécution de la séquence traduite, ce sont les règles standard de relance qui s'appliquent (voir le *Manuel de l'administrateur système*).
- Si une erreur se produit pendant la traduction de la séquence JCL, le système force le code de gravité d'erreur à 4. L'arrêt anormal du travail peut être évité au moyen d'un ordre JUMP avec une valeur de SEV appropriée (voir l'ordre JUMP).

#### 4.13.5 Paramètres

##### 4.13.5.1 Paramètre obligatoire

###### **description-fichier-d'entrée-séquentiel**

Identifie le fichier contenant la séquence JCL à exécuter. Le format de ce paramètre est détaillé au chapitre 3. Le fichier doit être en format DATA ou DATASSF.

##### 4.13.5.2 Paramètres facultatifs

###### **VALUES**

Ce paramètre indique les valeurs de paramètres de l'ordre EXECUTE à adopter pour la séquence JCL. Pour une description plus précise, se reporter à l'ordre INVOKE.

###### **PRTFILE**

"description-fichier-d'impression" identifie le fichier SYSOUT permanent destiné à contenir le compte rendu de traduction du JCL. Si ce paramètre est omis, l'état est envoyé dans un sous-fichier SYSOUT standard. "description-fichier-d'impression" a le même format que "description-fichier-d'entrée-séquentiel". Il est possible d'y rajouter les paramètres EXPDATE et DENSITY (voir l'ordre ASSIGN) ; de plus, la valeur implicite du paramètre ACCESS est WRITE.

###### **PRTDEF**

Ce paramètre permet d'indiquer des paramètres de mise en forme pour le fichier SYSOUT identifié par le paramètre PRTFILE. Les paramètres-define sont décrits dans l'ordre DEFINE.

###### **PRTOUT**

Ce paramètre permet de spécifier les options de traitement du fichier SYSOUT identifié par PRTFILE. N'importe quel paramètre de l'ordre SYSOUT (sauf "nom-fichier-interne") peut être spécifié (voir l'ordre SYSOUT).

### **SIZEOPT**

Ce paramètre permet de spécifier les caractéristiques de taille (paramètres-size) d'un espace d'exécution (voir l'ordre SIZE).

### **STEPOPT**

Ce paramètre permet de spécifier les options d'exécution de l'ordre EXECUTE. N'importe quel paramètre de l'ordre STEP peut être spécifié, à l'exception de DUMP=ALL, DEBUG et OPTIONS.

#### ***Exemple 1 :***

```
EXECUTE (EXEC.LIB, SUBFILE=SEQ1), VALUES=(X12, FILE=MY.LIB);
```

Cet ordre EXECUTE traduit, au moment de l'exécution de l'activité, la séquence JCL enregistrée dans l'unité SEQ1 de la bibliothèque EXEC.LIB en remplaçant chaque occurrence de &FILE par la valeur X12, et exécute les ordres traduits.

#### ***Exemple 2 :***

```
EXECUTE JCLSEQ.EXFL, STEPOPT=(DUMP=DATA, LINES=50);
```

Cet ordre EXECUTE traduit la séquence JCL enregistrée dans le fichier catalogué JCLSEQ.EXFL et soumet les ordres traduits pour exécution. Un maximum de 50 lignes peut être écrit par l'activité dans chaque fichier SYSOUT. En cas d'arrêt anormal de l'activité, le vidage ne doit concerner que les segments de données.

## 4.14 \$INPUT

### 4.14.1 Fonction

Ouvre une rubrique de description d'entrées et désigne le sous-fichier SYSIN dans lequel seront enregistrés les articles de la rubrique.

### 4.14.2 Rubrique

Description de travail.

### 4.14.3 Format

```
$INPUT nom-rubrique-de-description-d'entrées

TYPE=[ {DATA      }
      [ {COBOL    }
      [ {RPG      }
      [ {FORTRAN  }
      [ {JCL      }
      [ {COBOLX   }
      [ {GPL      }
      [ {DATASSF  }

[FORMAT=(chiffres3, chiffres3, chiffres3, chiffres3)]

[END={ENDINPUT  }
[    {DOLLAR   }
[    {EOF      }
[    {MATCH    }
[    {'chaîne8' }

[ENDCHAR='chaîne1']

[CONTCHAR='chaîne1']

[PRINT]      [CKSTAT]

[ {CVALUES}
[ {JVALUES}

[INFILE={ (nom-fichier-disquette
           { MEDIA={*                                } }
           { (nom-volume [nom-volume]...) } ) } ];
```

#### 4.14.4 Description

L'ordre \$INPUT et l'ordre \$ENDINPUT correspondant délimitent une rubrique de description d'entrées. Ce type de rubrique permet d'intégrer des données ou des commandes propres au processeur à une description de travail. Ces données servent d'entrée à un programme utilisateur ou à un utilitaire système, ou bien constituent un programme origine qui sert d'entrée à un compilateur (voir paramètre TYPE).

L'ordre \$INPUT est reconnu par le lecteur des entrées. Les données lues sont enregistrées dans un sous-fichier temporaire du fichier système SYSIN. Chaque sous-fichier SYSIN est identifié par le nom de rubrique de description d'entrées spécifié dans l'ordre \$INPUT.

Le symbole \$ est obligatoire à la première position de l'article.

Une rubrique de description d'entrées peut occuper n'importe quelle position dans une rubrique de description de travail ; elle ne peut cependant pas figurer à l'intérieur d'une rubrique de description d'activité. Il est par ailleurs possible de mettre une rubrique de description d'entrées à la disposition de plusieurs activités d'un même travail.

L'ordre ASSIGN portant sur une rubrique de description d'entrées se présente sous la forme :

```
ASSIGN nom-fichier-interne, *nom-rubrique-de-description-
d'entrées;
```

#### Commentaires et restrictions

- Les noms de rubriques de description d'entrées doivent être uniques à l'intérieur d'une rubrique de description de travail.
- Dans n'importe quelle activité, le nombre total des rubriques de description d'entrées et des ordres ASSIGN (y compris ceux portant sur des rubriques de description d'entrées) ne doit pas dépasser 31.
- Un maximum de 252 rubriques de description d'entrées en JCL (ou 251 en GCL) peuvent être associées à un travail donné. Noter qu'une seule définition de rubrique de description d'entrées (c'est-à-dire les articles de données entre \$INPUT et \$ENDINPUT) peut donner naissance à plusieurs rubriques de description d'entrées si le mécanisme de remplacement des paramètres est utilisé (voir plus loin).

#### Remplacement de paramètres dans les rubriques de description d'entrées

Une rubrique de description d'entrées peut contenir des références à des paramètres (sous la forme "&chaîne"), définies dans la description du travail en cours par un ou plusieurs ordres SET\_VALUES ou MODIFY\_VALUES. Le remplacement de paramètres ne peut avoir lieu que si le paramètre JVALUES est spécifié dans l'ordre \$INPUT associé.

Chaque fois qu'une référence à la rubrique de description d'entrées est rencontrée au cours de la traduction du JCL, une nouvelle version de la rubrique de description d'entrées est créée : chaque référence de paramètre est remplacée par la valeur qui lui est associée. La rubrique de description d'entrées ainsi mise à jour sera utilisée au moment de l'exécution.

Si CVALUES est spécifié dans \$INPUT, le système remplace les références aux paramètres par les valeurs indiquées ou implicites au niveau d'emboîtement en cours (voir l'ordre(INVOKE).

Si JVALUES est spécifié, les valeurs applicables au niveau de la description du travail (par exemple spécifiées dans un ordre RUN) sont utilisées, quel que soit le niveau d'emboîtement de l'ordre ASSIGN associé.

Le remplacement de paramètres est expliqué en détail dans le *Guide utilisateur JCL*.

### Restrictions

- Les rubriques de description d'entrées contenant des références à des paramètres peuvent être associées (par ASSIGN) à des séquences d'ordres JCL appelées par(INVOKE, mais le remplacement de paramètres est interdit à l'intérieur d'une rubrique de description d'entrées appelée par(INVOKE (voir l'ordre(INVOKE).
- Le remplacement de paramètres n'est autorisé qu'à l'intérieur des rubriques de description d'entrées de type DATA ou DATASSF.
- Une rubrique de description d'entrées contenant l'un ou l'autre des paramètres CVALUES ou JVALUES ne peut pas être utilisée plus de 31 fois dans des ordres ASSIGN d'activités, quel que soit le nombre de ces activités.

### Protection contre le remplacement de paramètres

Dans une rubrique de description d'entrées, une série d'articles consécutifs peut être "protégée" contre le mécanisme de remplacement de paramètres de la façon suivante : un article contenant les caractères "//BOD" (à partir de la colonne 1) doit précéder le premier article à protéger et un article contenant les caractères "//EOD" dans les cinq premières colonnes doit suivre le dernier article à protéger. Chaque fois qu'une version de la rubrique de description d'entrées est créée, ces deux articles de bornage sont retirés de la séquence, mais aucun remplacement de paramètre n'est effectué à l'intérieur de la séquence.

Cette possibilité est particulièrement utile lorsqu'une rubrique de description d'entrées contient des ordres de mise en forme de MAINTAIN\_LIBRARY dans lesquels les caractères & ne doivent pas être interprétés comme des références à des paramètres (pour toute précision sur les fonctions de mise en forme de MAINTAIN\_LIBRARY, voir le *Guide utilisateur de l'éditeur ligne*).

### Ordre d'application des paramètres

Les paramètres de l'ordre &INPUT s'appliquent dans l'ordre suivant :

#### CKSTAT

Si CKSTAT est spécifié, \$SWINPUT s'applique en premier lieu, sauf dans les deux cas suivants :

1. END=DOLLAR a été spécifié. Un ordre \$SWINPUT dans la rubrique de description d'entrées est alors interprété comme la fin de cette rubrique et non comme un ordre \$SWINPUT.
2. END='\$SWINPUT' ou END='\$SWI'a été spécifié. Un ordre \$SWINPUT ou \$SWI dans la rubrique de description d'entrées est interprété comme la fin de cette rubrique et non comme un ordre \$SWINPUT, sauf si l'ordre \$SWINPUT (ou \$SWI) et ses paramètres figurent dans le même article.

#### *Exemple :*

```
$SWINPUT XXXX END='$SWINPUT' ;
$SWINPUT FILE-A;
```

Le second \$SWINPUT est traité comme un ordre \$SWINPUT et non comme la fin de la rubrique de description d'entrées.

Par contre,

```
$SWINPUT XXXX END='$SWINPUT' ;
$SWINPUT
FILE-A;
```

le second \$SWINPUT est traité comme la fin de la rubrique de description d'entrées et non comme un ordre \$SWINPUT.

#### CONTCHAR et (ou) ENDCHAR

Cette fonction est appliquée ensuite.

#### FORMAT

Cette fonction est appliquée au résultat des deux fonctions précédentes.

#### Paramétrage

Le paramétrage de la rubrique de description d'entrées est appliqué en dernier.

## 4.14.5 Paramètres

### 4.14.5.1 Paramètre obligatoire

#### **nom-rubrique-de-description-d'entrées**

C'est le nom par lequel le programmeur désigne le sous-fichier SYSIN contenant les articles de la rubrique de description d'entrées. Ce nom ne doit pas dépasser 16 caractères et peut comporter des lettres, des chiffres (0 à 9) le trait d'union et le caractère de soulignement. "nom-rubrique-de-description-d'entrées" doit être unique à l'intérieur d'un travail.

### 4.14.5.2 Paramètres facultatifs

#### **TYPE**

Ce paramètre indique le type (et le format qui en découle) des articles de la rubrique de description d'entrées. Les types RPG, FORTRAN, JCL, COBOLX, GCL et GPL sont décrits (avec les formats correspondants) dans le *manuel de référence relatif à la maintenance des bibliothèques*. Les autres valeurs sont comme suit :

DATA	Données destinées à un programme utilisateur ou à un utilitaire système, introduites sans modification du format ni adjonction d'en-tête au début de chaque article. Cette valeur peut également être utilisée lorsque des instructions origine COBOL, FORTRAN ou RPG sont introduites. Si TYPE et FORMAT sont omis, DATA est implicite.
COBOL	Programme origine COBOL. Un en-tête SSF est ajouté au début de chaque article d'entrée et traité automatiquement par le système.
DATASSF	Un en-tête SSF est ajouté au début de chaque article d'entrée et traité automatiquement par le système. Chaque article de données est numéroté (dans l'en-tête SSF) à partir de 10, avec un pas de progression de 10. Un article de commande SSF est placé au début du sous-fichier SYSIN. Si FORMAT est spécifié mais TYPE omis, la valeur implicite de TYPE est DATASSF.

DATASSF peut être utilisée dans les cas suivants :

- lorsque les données d'entrée destinées à un programme résultant COBOL proviennent de SYSIN ;
- lorsque des instructions origine RPG sont introduites ;
- lorsque des instructions origine FORTRAN sont introduites.

**FORMAT**

Ce paramètre spécifie le format des articles d'entrée. Si FORMAT est spécifié, il ne faut pas spécifier TYPE=DATA pour les articles de sortie. Lorsque FORMAT est omis, sa valeur implicite est le format correspondant à TYPE (voir le *Manuel de référence relatif à la maintenance des bibliothèques*).

Les 4 valeurs spécifient respectivement :

- La position, dans l'article d'entrée, du premier chiffre du numéro de ligne. La première position d'un article est la position 1 et les espaces de gauche sont traités comme autant de zéros. "0" signifie qu'il n'y a pas de numéros de ligne.
- La position, dans l'article d'entrée, du dernier chiffre du numéro de ligne. "0" signifie qu'il n'y a pas de numéros de ligne.
- La position, dans l'article d'entrée, du premier caractère du texte.
- La position, dans l'article d'entrée, du dernier caractère du texte.

**END**

Ce paramètre indique au lecteur de travaux comment la rubrique de description d'entrées se termine :

ENDINPUT	Rubrique de description d'entrées terminée par le prochain ordre \$ENDINPUT. C'est la valeur implicite, si la rubrique n'est pas enregistrée dans un fichier disquette séparé.
DOLLAR	Rubrique de description d'entrées terminée par le prochain article d'entrée contenant le signe dollar (\$) en colonne 1. Cet article est considéré comme n'appartenant pas à la rubrique, sauf s'il s'agit de l'ordre \$HOL, \$BIN ou \$MAP.
EOF	Rubrique de description d'entrées terminée, dans un fichier disquette, par la condition "fin de fichier". Ce paramètre ne peut être spécifié que si INFILE spécifie un fichier sur disquette, auquel cas il constitue également la valeur implicite.
MATCH	Rubrique de description d'entrées terminée par le premier ordre \$ENDINPUT spécifiant son nom dans cet ordre. Ce paramètre permet l'insertion d'une rubrique de description d'entrées complète dans la rubrique de description d'entrées définie par cet ordre \$INPUT et l'ordre \$ENDINPUT correspondant.
chaîne8	La rubrique de description d'entrées se termine au premier article commençant par cette chaîne de caractères. Cet article de terminaison n'est pas traité comme un article de données de la rubrique. Il doit être codé en EBCDIC.

Lorsque le paramètre END est omis, la rubrique de description d'entrées est close par l'ordre \$ENDINPUT ou par la condition fin de fichier.

**ENDCHAR**

Ce paramètre s'applique uniquement aux articles de rubrique de description d'entrées du type DATA ou DATASSF. Il peut s'utiliser pour les rubriques de description d'entrées dont les articles contiennent tous un caractère donné comme dernier caractère avant les espaces fin. Ce caractère est spécifié dans le paramètre ENDCHAR. Il sera enlevé, avec tous les espaces fin, des articles de la rubrique de description d'entrées. La valeur fournie par ENDCHAR peut être protégée (encadrée par des apostrophes). Noter qu'il est possible de spécifier un espace comme dernier caractère, mais il doit alors être protégé.

Lorsque ENDCHAR est mentionné, tout article ne contenant pas le dernier caractère indiqué à la position requise est concaténé avec l'article suivant. La concaténation se poursuit à concurrence de la longueur maximum d'article (256 caractères), jusqu'à détection d'un article contenant le caractère indiqué à la dernière position avant les espaces fin. ENDCHAR permet donc à l'utilisateur de spécifier un caractère donné pour indiquer la fin d'article.

**ATTENTION**

Si ENDCHAR apparaît dans \$INPUT, le dernier article doit obligatoirement contenir le caractère spécifié.

**Exemples d'utilisation de ENDCHAR**

Dans les exemples qui suivent, le caractère "b" représente un espace.

***Exemple 1 :***

ENDCHAR=Q

Article introduit (80 caractères) :

1234Q5678Qbb9bQbbb.....b

Article enregistré (14 caractères) :

1234Q5678Qbb9b

***Exemple 2 :***

ENDCHAR= ' b '

Article introduit (80 caractères) :

bVENIBVIDIBVICIbbb.....b

Article enregistré (15 caractères) :

bVENIBVIDIBVICI

**Exemple 3 :**

ENDCHAR=/

**Articles introduits (80 caractères) :**

AAAbbb.....b  
BBB/bbb.....b  
CCCbbb.....b  
DDD/bbb.....b  
EEEbbb.....b

**Articles enregistrés:**

AAAbbb .....b BBB (83 caractères)  
CCCbbb.....b DDD (83 caractères)  
EEE (3 caractères)

**CONTCHAR**

Ce paramètre ne s'applique qu'aux articles de type DATA ou DATSSF. Pour tout article introduit contenant le caractère indiqué à la dernière position avant les espaces fin, ce caractère est retiré, ainsi que les espaces fin, et l'article est concaténé avec le suivant. Le caractère suite peut être protégé par des apostrophes. Un article formé par utilisation de CONTCHAR ne doit pas dépasser 255 caractères.

**Exemple d'utilisation de CONTCHAR**

Dans l'exemple qui suit, le caractère b représente un espace.

CONTCHAR=-

**Articles introduits :**

ABCDE-FFF-bbbAXZ-bbb.....b  
12345-678-bbb999bbb.....b  
-XYZbbb.....b

**Articles enregistrés**

ABCDE-FFF-bbbAXZ12345-678bbb999bbb.....b  
-XYZbbb.....b

### **Exemples d'utilisation combinée de ENDCHAR et CONTCHAR**

Les paramètres ENDCHAR et CONTCHAR peuvent être combinés dans le même ordre \$INPUT, à condition d'indiquer pour chacun d'entre eux un caractère différent.

Dans les exemples qui suivent, le caractère b représente un espace.

#### ***Exemple 1 :***

CONTCHAR=-, ENDCHAR=9

Articles introduits :

ABCD9b-bXZ-bbb.....b  
XZZZ9b-bAB-9bbb.....b

Articles enregistrés :

ABCD9b-bXZXZZ9b-bAB-

#### ***Exemple 2 :***

Soit la rubrique de description d'entrées suivante (80 caractères par article) :

```
$INPUT, CONTCHAR=+, ENDCHAR=/;  
AAAbbb.....b  
BBB/bbb.....b  
CCCbbb.....b  
DDD+bbbb.....b  
EEE/bbb .....b  
FFFbbb .....b  
GGG+bbbb .....b  
HHH/bbb .....b  
$ENDINPUT;
```

Les articles suivants sont enregistrés :

AAAbbb .....b BBB (83 caractères)  
CCCbbb.....b DDDEEE (86 caractères)  
FFFbbb .....b GGGHHH (86 caractères)

#### **ATTENTION**

Si le résultat de l'utilisation de ENDCHAR et/ou CONTCHAR est un article de longueur zéro, cet article n'est pas pris en compte.

### **PRINT**

Lorsque ce paramètre est spécifié, le contenu de la rubrique de description d'entrées est consigné dans la partie traduction du JOR au moment de l'introduction du travail. Il est possible d'y faire figurer jusqu'à 200 articles d'entrée par travail.

**CKSTAT**

Lorsque ce paramètre est spécifié, le lecteur des entrées recherche un ordre \$SWINPUT dans chaque article de la rubrique de description d'entrées. S'il en trouve un, il l'interprète et l'exécute.

**CVALUES**

Ce paramètre ne s'applique qu'aux articles d'entrée de type DATA ou DATASSF. Il indique :

- que la rubrique de description d'entrées contient des références à des paramètres ;
- qu'au moment de l'exécution, le système utilisera une rubrique mise à jour par remplacement de valeurs de paramètres, et que les valeurs de substitution seront celles qui s'appliquent au niveau en cours du travail.

**JVALUES**

Ce paramètre ne s'applique qu'aux articles d'entrée de type DATA ou DATASSF. Il indique :

- que la rubrique de description d'entrées contient des références à des paramètres ;
- qu'au moment de l'exécution, le système utilisera une rubrique mise à jour par remplacement de valeurs de paramètres, qui s'appliquent au niveau du travail.

**INFILE**

Ce paramètre est valable uniquement lorsque la description de travail est enregistrée sur disquette. Il indique où se trouvent les articles d'entrée.

nom-fichier-disquette	Nom externe du fichier disquette contenant les articles d'entrée.
MEDIA=*	Aucun nom de volume disquette n'est spécifié. C'est le premier volume contenant "nom-fichier-disquette" qui est retenu. Le fichier concerné peut être multivolume.
MEDIA=(nom-volume)	Le fichier doit être recherché sur le(s) volume(s) spécifié(s).

Lorsque MEDIA est omis, le fichier est recherché sur le volume disquette contenant la description du travail.

Le paramètre **INFILE** permet de retrouver une rubrique de description d'entrées dans des fichiers disquette qui peuvent être dans le même volume que la description du travail, ou dans les volumes suivants du train d'entrées. Lorsqu'une rubrique de description d'entrées est enregistrée dans un fichier disquette multivolume, seuls les volumes contenant le premier ou le dernier fragment peuvent contenir des descriptions de travaux ou d'autres fichiers avec rubriques de description d'entrées. Ainsi, les seules données extraites des volumes intermédiaires (ceux qui ne contiennent ni le premier, ni le dernier fragment) d'un fichier disquette multivolume, proviennent des fragments de ce fichier, aucun autre fichier de ces volumes n'étant accessible pendant le processus. Une restriction similaire s'applique à la concaténation de fichiers disquette multifragments (voir ce qui concerne la concaténation de fichiers dans la description de l'ordre **ASSIGN**).

**Exemple 1 :**

```
$JOB MYJOB, USER=SMITH, PROJECT=Q141;
STEP MYPROG, (NEW.MYLIB, DEVCLASS=MS/M500, MEDIA=B015);
ASSIGN FILE1, *MYFILE;
SYSOUT PRTFILE;
ENDSTEP;
$INPUT MYFILE;
.
.
.
data
.
.
.
$ENDINPUT;
$ENDJOB;
```

Dans cet exemple, la rubrique de description d'entrées **MYFILE** contient des données et se termine par l'ordre **\$ENDINPUT**.

**Exemple 2 :**

Si le travail de l'exemple ci-dessus avait été lu à partir d'une disquette, l'ordre **\$INPUT** aurait pu être formulé comme suit :

```
$INPUT MYFILE INFILE=(FILE1, MEDIA=(XYZ1, XYZ2, XYZ3));
```

Les données de la rubrique de description d'entrées seront extraites du fichier **FILE1**, qui sera recherché dans les volumes **XYZ1**, **XYZ2** et **XYZ3**. La rubrique de description d'entrées sera close à la lecture d'un ordre **\$ENDINPUT** ou sur détection d'une fin de fichier sur **FILE1**.

## 4.15 INVOKE

### 4.15.1 Fonction

Insère dans la description de travail en cours une séquence d'ordres JCL en format origine, se trouvant soit dans une unité de bibliothèque, soit dans une rubrique de description d'entrées.

### 4.15.2 Rubrique

Description de travail.

### 4.15.3 Format

```
INVOKE
{ *nom-rubrique-de-description-d'entrées1
  {nom-unité-bibliothèque [{(description-bibliothèque-d'entrée)}]} }
  [{SYS}]}}

VALUES=[valeur-paramètre1 [valeur-paramètre2]...]
      [mot-clé1=valeur-mot-clé1 [mot-clé2=valeur-mot-clé2]...])

[UPDATE=*nom-rubrique-de-description-d'entrées2]

[LIST=ALL];
```

### 4.15.4 Description

L'ordre INVOKE demande l'introduction d'une séquence d'ordres JCL en format origine à partir d'une bibliothèque origine d'ordres JCL ou d'une rubrique de description d'entrées.

Si une étiquette est définie à l'intérieur d'une séquence appelée par INVOKE, elle ne peut être citée à l'extérieur de cette séquence.

Il est par ailleurs impossible d'effectuer des branchements par JUMP à l'extérieur de la séquence appelée.

Noter que la même étiquette peut être définie à l'intérieur et à l'extérieur de la séquence sans risque d'ambiguïté.

Enfin, un ordre LIB figurant dans une séquence appelée s'applique uniquement à cette séquence.

Une séquence d'ordres JCL appelée par INVOKE peut contenir d'autres ordres INVOKE : jusqu'à neuf niveaux d'emboîtement sont ainsi autorisés, et ces séquences appelées peuvent être enregistrées dans quinze bibliothèques maximum.

Une séquence d'ordres appelés par INVOKE doit contenir une ou plusieurs rubriques de description d'activité entières, mais ne doit comporter ni rubrique de description d'entrées, ni ordre \$JOB ou \$ENDJOB. Le paramètre VALUES permet le remplacement de paramètres dans la séquence appelée et le paramètre UPDATE permet la mise en forme de la séquence appelée avant traduction.

Si la séquence appelée contient l'affectation par ASSIGN d'une rubrique de description d'entrées dont l'ordre \$INPUT spécifie le paramètre CVALUES, une nouvelle rubrique de description d'entrées sera créée par remplacement des valeurs de paramètres alors applicables à la séquence appelée (voir l'ordre \$INPUT).

Les fichiers temporaires et permanents inscrits dans des catalogues privés ne doivent pas être utilisés, sauf si le catalogue concerné est automatiquement associable ou si le train d'entrées est soumis pour exécution au moyen d'un ordre RUN.

## 4.15.5 Paramètres

### 4.15.5.1 Paramètres obligatoires

#### **nom-rubrique-de-description-d'entrées1**

C'est le nom de la rubrique de description d'entrées SYSIN qui contient la séquence d'ordres JCL appelée, en format DATA ou DATASSF. L'ordre \$INPUT correspondant ne doit spécifier ni CVALUES, ni JVALUES, le remplacement de paramètres n'étant pas autorisé à l'intérieur d'une rubrique de description d'entrées appelée.

#### **nom-unité-bibliothèque**

Identifie le sous-fichier de bibliothèque contenant les ordres JCL à introduire. Ce nom ne doit pas dépasser 31 caractères (lettres, chiffres, trait d'union, caractère de soulignement) et doit commencer par une lettre ou un chiffre.

### 4.15.5.2 Paramètres facultatifs

#### **description-bibliothèque-d'entrée**

Identifie la bibliothèque utilisateur origine contenant l'unité de bibliothèque spécifiée. Voir au chapitre 3 le format de ce paramètre.

**ATTENTION**

Si le fichier bibliothèque n'est pas résident (s'il n'est pas implanté sur le disque système résident), il peut arriver que le système ne puisse pas y accéder au moment de la traduction des ordres JCL (par exemple, si le volume contenant le fichier n'est pas monté). Le travail est alors arrêté prématurément. Il est possible d'éviter cette situation en utilisant un ordre EXECUTE (voir le guide utilisateur JCL).

**SYS**

Ce paramètre indique la bibliothèque système résidente SYS.HSLLIB. En l'absence de description de bibliothèque d'entrée, SYS est implicite.

**VALUES**

La valeur de paramètre ou la valeur de mot-clé définit la valeur réelle à utiliser dans la séquence d'ordres JCL appelée à la place de la référence "&i" (1<i<99) ou "&mot-clé". Cette valeur peut être une chaîne protégée (n'importe quelle suite de caractères encadrée par des apostrophes et si une apostrophe apparaît à l'intérieur de la chaîne, elle doit être doublée) de 128 caractères maximum. Dans une chaîne, il est possible d'omettre une valeur de paramètre à condition de la remplacer par une virgule, ce qui donne deux virgules consécutives ou séparées par un espace ou plus.

Si des espaces sont utilisés comme séparateurs, les valeurs de paramètres manquantes peuvent être signalées par autant de caractères #. Dans ce cas, la valeur correspondante est une chaîne vide. Toutefois, s'il existe un ordre VALUES ou MODVL dans la séquence JCL appelée, il spécifie la valeur implicite du paramètre omis.

Si une "valeur-paramètre" de VALUES est NIL, la référence "&i" ou "&mot-clé" n'est pas remplacée et le paramètre correspondant de la séquence d'ordres JCL appelée n'est pas pris en compte (même si une valeur implicite est indiquée dans un ordre VALUES ou MODVL). Noter que "&0" et "&00" sont toujours remplacés par des chaînes vides.

**Exemple :**

Soit un ordre INVOKE ayant la forme suivante :

```
INVOKE SEQ4, SYS, VALUES=(NEW.MYFILE1, NIL, TAPES=3);
```

Si l'unité de bibliothèque SEQ4 contient :

```
ASSIGN TESTFILE, NEW.&1, &2, MOUNT=&TAPES;
.....
VALUES OLD.FIL, 'DEVCLASS=MS/M500, MEDIA=VOLA', TEMPRY;
.....
ASSIGN FIL2, &1, &2, &3;
```

le résultat sera le suivant :

```
ASSIGN TESTFILE, NEW.MYFILE1, MOUNT=3;
.....
VALUES OLD.FIL, 'DEVCLASS=MS/500, MEDIA=VOLA', TEMPRY;
.....
ASSIGN FIL2, MYFILE1, DEVCLASS=MS/M500, MEDIA=VOLA;
```

Les points suivants sont à noter :

- La valeur de &2 dans le premier ordre ASSIGN devient une suite vide.
- Pour le deuxième ordre ASSIGN, les valeurs de &1 et &3 de l'ordre INVOKE ("MYFILE1" et "NIL") l'emportent sur les valeurs indiquées dans l'ordre VALUES (OLD.FIL et TEMPRY).
- Le fichier portant le nom externe NEW.MYFILE1 dans le premier ordre ASSIGN peut être catalogué, transmis par une autre activité ou résidant, alors que le fichier correspondant du deuxième ordre ASSIGN est non catalogué.

## UPDATE

Ce paramètre indique que les commandes du mini-éditeur enregistrées dans la rubrique de description d'entrées "\*nom-rubrique-de-description-d'entrées2" doivent être appliquées à la séquence d'ordres JCL appelée avant traduction. L'utilisation du mini-éditeur est expliquée dans le guide utilisateur JCL

L'ordre \$INPUT correspondant ne doit contenir ni le paramètre CVALUES, ni le paramètre JVALUES (remplacement de paramètres interdit à l'intérieur de la rubrique de description d'entrées). Noter que la séquence d'ordres JCL appelée peut être elle-même une rubrique de description d'entrées.

### *Exemple 1 :*

```
INVOKE SEQ1, INV.LIB, VALUES=(X12, FILE=MY.LIB),
      UPDATE=*UP;
```

Le contenu de l'unité SEQ1 de la bibliothèque INV.LIB est mis à jour conformément aux commandes d'édition figurant dans la rubrique de description d'entrées UP. Les remplacements de paramètres par les valeurs de VALUES sont ensuite effectuées sur SEQ1 et la version finale est insérée dans la description de travail en cours.

### *Exemple 2 :*

```
INVOKE *CRDSEQ, UPDATE=*EDITS;
```

Le contenu de la rubrique de description d'entrées CRDSEQ est mis en forme conformément aux commandes de la rubrique de description d'entrées EDITS, et le résultat est inséré dans la description de travail en cours.

## LIST=ALL

Le JCL origine et tous les développements de séquences JCL sont listés, ainsi que les messages d'erreur.

## 4.16 \$JOB

### 4.16.1 Fonction

Marque le début d'une rubrique de description de travail et fournit l'identification du travail.

### 4.16.2 Rubrique

Description de travail.

### 4.16.3 Format

```
$JOB nom-travail
[USER=nom-utilisateur]
[PROJECT=nom-projet]
[BILLING=nom-compte]
[NSTARTUP]
[ {SOURCE} ]
[LIST={NO {ALL}}]
[ {NORMAL} ]
[JOR={NO {ABORT}}]
[ {identificateur2} ]
[CLASS={P {}}]
[ {HOLD {}} ]
[ {HOLD=chiffres2} ]
[HOLDOUT]
[PRIORITY=chiffre1]
[ {110 {}} ]
[RECSIZE={chiffres3}]
[REPEAT]
[HOST = {nom4}]
[JOBLANG = {GCL {JCL}}]
[EXPVAL];
```

#### 4.16.4 Description

S'il est utilisé, \$JOB doit être le premier ordre d'une description de travail. Son rôle est principalement d'identifier un travail en spécifiant son nom d'utilisateur, son nom de projet et un nom de travail symbolique. Les ordres \$JOB et \$ENDJOB peuvent être omis lorsqu'un seul travail est soumis au moyen d'un ordre RUN ou d'une commande opérateur EJR.

Le lecteur des entrées reconnaît l'ordre \$JOB et range les ordres de description du travail en mémoire de réserve.

Le symbole \$ doit obligatoirement figurer à la première position de l'article, sauf lorsque le travail est soumis au moyen d'un ordre RUN ou d'une commande opérateur EJR.

Toute valeur mentionnée dans l'ordre \$JOB pour USER, PROJECT et BILLING doit correspondre exactement aux rubriques du catalogue de site (si ce catalogue existe et est valide). Le compte d'une commande EJR n'est significatif que si le travail soumis ne contient pas l'instruction \$JOB.

#### 4.16.5 Paramètres

##### 4.16.5.1 Paramètres obligatoires

###### **nom-travail**

Nom sous lequel l'utilisateur connaît le travail. Il peut comporter jusqu'à huit caractères (lettres, chiffres, trait d'union et caractère de soulignement), et doit commencer par une lettre ou un chiffre. Lorsque "nom-travail" comporte plus de 8 caractères, seuls les 8 premiers sont retenus, sauf s'il fait plus de 16 caractères, auquel cas le travail est arrêté prématurément.

Le nom de travail doit figurer dans le même article que le nom de l'ordre (\$JOB). Noter que le système connaît le travail sous son numéro de passage (RON).

En l'absence d'ordre \$JOB, le nom de travail est le nom d'unité de bibliothèque (RUN unité-bibliothèque ou EJR) ou le nom de fichier externe (RUN nfe).

###### **USER**

Ce paramètre indique le nom de l'utilisateur. Il peut comporter jusqu'à 12 caractères (lettres, chiffres, trait d'union et caractère de soulignement), dont le premier doit être une lettre ou un chiffre.

Le paramètre USER est facultatif si le travail est soumis de l'une des trois façons suivantes :

- au moyen d'une commande opérateur EJR ; c'est alors l'identification du soumettant qui est implicitement adoptée ;
- au moyen d'un ordre RUN ; c'est alors l'utilisateur IOF ou le demandeur de la génération dynamique qui est implicitement retenu.

En dehors de ces deux exceptions, si le paramètre USER est omis, l'exécution du travail n'est pas demandée. Le travail reçoit simplement un numéro de passage (RON) et le JOR est généré.

Lorsque des droits d'accès ont été définis, la valeur de USER peut être limitée comme suit :

- Si l'utilisateur soumettant le travail est l'opérateur principal ou un opérateur de terminal, il est possible de spécifier n'importe quelle valeur pour USER.
- Si l'utilisateur a soumis le travail en traitement par lots ou sous IOF, c'est son nom qui doit être spécifié dans USER.

#### 4.16.5.2 Paramètres facultatifs

##### **PROJECT**

Ce paramètre indique le nom du projet associé à l'utilisateur spécifié. Ce nom peut comporter jusqu'à 12 caractères (lettres, chiffres, trait d'union et caractère de soulignement), dont le premier doit être une lettre ou un chiffre.

##### **ATTENTION**

Le paramètre PROJECT est obligatoire dans l'ordre \$JOB si aucune valeur implicite n'a été indiquée dans le catalogue de site ou s'il n'existe pas de catalogue de site.

##### **BILLING**

Ce paramètre indique le nom utilisé pour l'établissement de statistiques sur le projet en cours et son contrôle. Ce nom peut comporter jusqu'à 12 caractères (lettres, chiffres, trait d'union et caractère de soulignement), dont le premier doit être une lettre ou un chiffre. Lorsque BILLING est omis et qu'aucune valeur implicite n'a été définie dans le catalogue de site associé au projet en cours, c'est le nom de projet qui est utilisé comme nom de compte dans la bannière du JOR.

**ATTENTION**

Le paramètre BILLING est obligatoire dans l'ordre \$JOB si aucune valeur implicite n'a été indiquée dans le catalogue de site ou s'il n'existe pas de catalogue de site.

**NSTARTUP**

Si elles existent dans la bibliothèque origine de lancement SITE.STARTUP, les séquences de lancement sont insérées automatiquement après l'ordre \$JOB. Une séquence de lancement peut comprendre l'ensemble des commandes et des directives système exécutables en mode traitement par lots.

En traitement par lots, il est possible d'associer deux séquences de lancement à un projet dans le catalogue : une séquence obligatoire et une séquence facultative.

Il existe trois séquences de lancement :

SITE\_B  
<projet>\_B  
<projet>\_<utilisateur>\_B

**Règles de mise en oeuvre des séquences de lancement :**

1. N'importe laquelle de ces trois séquences peut être déclarée obligatoire.
2. Seules  
<projet>\_B  
et  
<projet>\_<utilisateur>\_B  
peuvent être déclarées facultatives.
3. La séquence obligatoire est exécutée avant la séquence facultative.
4. L'insertion automatique de la séquence facultative peut être invalidée en spécifiant NSTARTUP dans \$JOB, STARTUP =0 dans EJR ou dans RUN\_JOB, ou en recourant à la zone STARTUP de la structure d'entrée en cas d'utilisation de l'interface de programmation.
5. Pour un projet SYSADMIN, si NSTARTUP est spécifié, l'insertion de la séquence obligatoire et celle de la séquence facultative sont toutes deux invalidées.

**LIST**

Ce paramètre permet de sélectionner le type d'informations devant figurer dans le listage des ordres JCL du JOR.

<b>SOURCE</b>	Ordres JCL origine, ordres insérés du lecteur des entrées, articles insérés au moyen d'un ordre \$SWINPUT spécifiant le paramètre CONSOLE et messages d'erreur.
---------------	---

ALL	Ordres JCL origine, tous les développements des séquences JCL introduites par <b>Invoke</b> et <b>\$SWINPUT</b> , séquence de lancement du projet et messages d'erreur.
NO	Ordres du lecteur des entrées, commentaires et messages d'erreur.

Lorsque des droits d'accès ont été définis et que l'utilisateur n'a pas le droit de lecture (READ) sur le ou les fichiers contenant les données devant figurer dans le JOR, LIST n'est pas pris en compte.

JOR

Ce paramètre détermine dans quels cas le JOR doit être généré. Les valeurs pouvant être spécifiées sont :

NORMAL	Le JOR est généré. C'est la valeur implicite.
ABORT	Le JOR est généré uniquement en cas d'arrêt prématuré du travail.
NO	Le JOR n'est pas généré.

## CLASS

Ce paramètre rattache le travail à une classe de travaux définie par une ou deux lettres. Il existe 16 classes à une lettre (de A à P) et 416 classes à deux lettres (de AA à PZ). La classe de travaux permet de définir les priorités implicites de planification et d'exécution, ainsi que la charge maximum (niveau de multiprogrammation) d'une classe, attributs s'appliquant à tous les travaux de la classe. L'opérateur peut suspendre et relancer une classe donnée, d'où une grande souplesse de traitement et une utilisation optimale des ressources de la machine. L'utilisation des classes de travaux permet en outre de contrôler la sélection des travaux et de gérer leur exécution en série.

Pour un travail donné, les possibilités de rattachement à une classe peuvent être limitées par la description du projet auquel il appartient dans le catalogue de site. Il est possible d'associer à chaque projet une classe IOF et une classe de traitement par lots implicites. Chaque projet peut être rattaché à 26 classes maximum. Tout travail soumis avec une classe qui n'existe pas ou n'est pas accessible au projet est rattaché à la classe P.

**HOLD**

Ce paramètre indique que le travail doit être bloqué (état HOLD) après traduction du JCL. Un travail bloqué ne peut pas être planifié pour exécution tant que l'opérateur n'a pas émis une commande RELEASE\_JOB (RJ) ou qu'un travail en exécution n'a pas supprimé l'état HOLD par un ordre RELEASE.

HOLD spécifié avec une valeur est un compteur, décrémenté de 1 à chaque ordre JCL "RELEASE". Le travail devient planifiable lorsque le compteur est à zéro. Il peut être directement forcé à zéro par une commande opérateur RJ avec l'option STRONG.

## **HOLDOUT**

Lorsque ce paramètre est spécifié, les fichiers de sortie générés par le travail ne sont pas automatiquement imprimés, mais retenus jusqu'à l'émission d'une commande opérateur RELEASE\_OUTPUT (RO).

## **PRIORITY**

Ce paramètre indique une priorité de planification d'un chiffre, comprise entre 0, priorité la plus élevée, et 7, priorité la plus faible. Les travaux sont planifiés en fonction de cette priorité.

La possibilité de définir une priorité de planification pour un travail peut être limitée par la description du projet auquel il est rattaché, dans le catalogue de site. Lorsque la priorité définie par l'utilisateur dépasse la valeur maximum autorisée dans le cadre du projet, c'est cette valeur maximum qui est adoptée, et un message d'avertissement est émis.

A l'intérieur d'une priorité donnée, les travaux sont planifiés suivant la règle du "premier entré-premier sorti" (FIFO).

**Remarque :** Un travail ne peut être planifié si le nombre maximum de travaux de sa classe a déjà été planifié.

La valeur implicite de PRIORITY dépend de la classe à laquelle appartient le travail (voir tableau 1-1 dans le chapitre 1).

## **RECSIZE**

Ce paramètre n'est spécifié que lorsque les entrées proviennent d'un terminal KDS. Il spécifie la longueur des articles pour les fichiers disquette transmis. La valeur implicite est 110.

## **REPEAT**

Ce paramètre indique que le travail peut être relancé depuis le début lorsque l'opérateur le souhaite, après une relance à chaud. Il n'a de sens que si l'activité au cours de laquelle s'est produit l'arrêt prématuré ne comporte pas elle-même de paramètre REPEAT.

Si une activité comporte un paramètre REPEAT, elle est relancée lorsque l'opérateur le décide. Si elle ne comporte pas de paramètre REPEAT, c'est le travail dans sa totalité qui est relancé.

Lorsque le paramètre REPEAT est omis, le travail ne peut pas être relancé entièrement.

## **HOST**

Identifie le site d'exécution du travail, s'il est différent de celui sur lequel le JCL est introduit ou enregistré. Le nom d'hôte comporte quatre caractères alphanumériques au maximum. Le site spécifié doit correspondre à un système DPS 7 ou DPS 7000.

### **JOBLANG**

Identifie le langage de commande utilisé pour décrire le travail :

GCL      Langage de commande GCOS

JCL      Langage de gestion des travaux

En cas d'omission, c'est la valeur entrée au moment de la soumission qui est, le cas échéant, adoptée. Si cette valeur a également été omise, le paramètre JOBLANG prend implicitement la valeur JCL.

### **EXPVAL**

Une autre liste des ordres JCL avec les valeurs développées des paramètres doit être imprimée dans l'historique du travail (JOR) lors de la traduction du JCL. Ce paramètre ne s'applique que si JOBLANG = JCL.

*Exemple :*

```
$JOB DMJOB, USER=DJM, PROJECT=MECTP, HOST=BP3C;
```

Dans l'ordre ci-dessus, le travail DMJOB sera transféré vers le système BP3C, puis exécuté sur ce système.

## 4.17 JUMP

### 4.17.1 Fonction

Modifie, sous certaines conditions ou dans tous les cas, l'ordre d'exécution (normalement séquentiel) des ordres JCL. Permet également d'éviter l'arrêt prématuré d'un travail en cas d'arrêt anormal d'une activité.

### 4.17.2 Rubrique

Description de travail ou d'activité.

### 4.17.3 Format

```

JUMP {nom-étiquette}
      {CONTINUE      }

[ {      }      {EQ}      ]
[ {STATUS}      {NE}      ]
[ {SEV}        {GE}      ]
[ {SWi}        {GT}      ]      chiffres5];
[ {IOF}        {LE}      ]
[ {      }      {LT}      ]

```

### 4.17.4 Description

L'ordre JUMP modifie l'ordre séquentiel d'exécution des ordres JCL, sous certaines conditions ou dans tous les cas. Le traitement passe normalement de l'ordre en cours à l'ordre JCL portant l'étiquette correspondant au nom d'étiquette spécifié dans l'ordre JUMP. Si le paramètre CONTINUE est spécifié, il y a passage à l'ordre JCL suivant, ce qui est particulièrement utile en cas d'arrêt prématuré d'une activité (voir ci-après).

L'ordre JUMP peut tester le code d'état ou le code de gravité d'erreur résultant d'une activité, ou encore un inverseur système antérieurement positionné par l'utilisateur au moyen d'un ordre LET ou par un programme COBOL (inverseur logique externe) ; le branchement a lieu ou non selon la valeur en cours du code ou de l'inverseur testé.

Si "nom-étiquette" (ou CONTINUE) est seul spécifié, il y a branchement inconditionnel.

Si JUMP figure dans une rubrique de description d'activité, l'ordre JCL correspondant d'après son étiquette doit figurer dans la même rubrique de description d'activité.

La spécification de JUMP dans une rubrique de description d'activité permet de sauter certains ordres JCL de cette rubrique lorsqu'une condition testée est vérifiée.

Lorsque JUMP figure dans une rubrique de description de travail, mais en dehors de toute rubrique de description d'activité, l'ordre JCL dont l'étiquette est spécifiée doit figurer dans la même rubrique de description de travail, également en dehors de toute rubrique de description d'activité.

Un branchement à l'intérieur d'une rubrique de description de travail peut s'effectuer vers l'amont ou vers l'aval (c'est-à-dire sur un ordre qui précède ou suit JUMP). Il est ainsi possible de sauter ou réexécuter successivement autant d'activités que nécessaire. A l'intérieur d'une rubrique de description d'activité, un branchement ne peut s'effectuer que vers l'aval.

L'ordre JUMP ne prenant effet qu'à l'exécution du travail, un branchement sur un ordre du traducteur JCL (par exemple, INVOKE ou VALUES) n'a aucun sens.

Noter également que l'ordre JUMP ne peut pas être utilisé pour sauter un ordre LIB, qui lui, est traité lors de la phase traduction (voir l'exemple 4 ci-après). Cette remarque s'applique également aux ordres VALUES et MODVL (voir l'exemple 5 ci-après).

### **STATUS, SEV, SWi, IOF**

Un branchement conditionnel n'a lieu que si la condition testée est vérifiée. Si une activité s'arrête prématurément, il en est de même pour le reste du travail, sauf si un ordre JUMP testant la variable système STATUS ou SEV suit les ordres JCL de description de l'activité (autrement dit, si JUMP suit l'ordre ENDSTEP de l'activité arrêtée et précède l'éventuel ordre STEP suivant). Dans ce cas, si la condition est vérifiée, l'exécution normale du travail se poursuit par l'ordre qui porte l'étiquette indiquée. Tout ordre JCL figurant entre le point auquel l'activité s'est arrêtée et l'ordre JUMP est sauté. Si c'est le mot réservé CONTINUE qui est spécifié comme étiquette, l'exécution passe à l'ordre JCL suivant.

## **4.17.5 Paramètres**

Les paramètres de l'ordre JUMP sont positionnels (c'est-à-dire qu'il faut les spécifier dans l'ordre indiqué par le format).

### **4.17.5.1 Paramètres obligatoires**

#### **nom-étiquette**

C'est l'étiquette affectée au prochain ordre JCL à exécuter, sur condition ou dans tous les cas. Ce nom ne doit pas dépasser huit caractères (lettres, chiffres, trait d'union et caractère de soulignement), et doit commencer par une lettre ou un chiffre.

#### **CONTINUE**

Il s'agit d'un nom réservé indiquant que même en cas d'arrêt anormal d'activité, l'exécution du travail doit se poursuivre par l'ordre suivant.

## 4.17.5.2 Paramètres facultatifs

**STATUS**

Ce paramètre sert à tester le code fin d'activité positionné par un programme. La valeur numérique ("chiffres5") qui suit l'opérateur de relation indiqué (par exemple EQ) est initialisée à 0 au moment du lancement de l'activité. Les valeurs possibles sont les suivantes :

- Arrêt normal ou arrêt prématuré déclenché par l'utilisateur (paramètre REPEAT non mentionné dans STEP) :  
l'utilisateur indique une valeur entre 0 (normal) et 32767. Une valeur égale ou supérieure à 10000 représente un état d'arrêt prématuré.
- Arrêt prématuré déclenché par l'opérateur : 50000.
- Arrêt prématuré déclenché par le système : valeur égale ou supérieure à 60000.

**Remarque :** 0 indique une fin normale.

**SEV**

Ce paramètre sert à tester le code de gravité d'erreur d'une activité forcé soit par un programme utilisateur, soit par un arrêt prématuré du système, soit par une commande opérateur TERMINATE\_JOB (TJ). La valeur numérique "chiffres5" peut être un nombre entier de 0 à 6 inclus. Le tableau suivant montre la correspondance entre SEV et STATUS :

SEV	STATUS
0	0-99
1	100-999
2	100-9999
3	10000-19999
4	20000-32767
5	50000
6	> 60000

**SWi**

Ce paramètre indique qu'un inverseur système a été positionné par l'utilisateur au moyen d'un ordre LET, le positionnement à la valeur 0 ou 1 étant associé à une condition du programme.

Le caractère i représente un nombre entier compris entre 0 et 31, qui indique l'inverseur logique à tester.

La valeur numérique "chiffres5" peut prendre la valeur 0 ou 1.

Les inverseurs logiques sont initialisés à la valeur 0 au moment du lancement du travail (ordre \$JOB). Ils peuvent être initialisés à une autre valeur pour un travail généré dynamiquement (ordre RUN), ou pour un travail à l'état HOLD remis en planification (ordre RELEASE).

### IOF

Ce paramètre est un inverseur système. "chiffres5" est positionné à 1 si le travail est exécuté en mode IOF, et à 0 si le travail est exécuté en traitement par lots.

### Opérateurs de relation

Ils ont la signification suivante :

EQ	Egal à
NE	Différent de
GE	Supérieur ou égal à
GT	Supérieur à
LE	Inférieur ou égal à
LT	Inférieur à

### Exemple 1 :

```
$JOB A, ....;
STEP LM1, ....;
.
.
.
ENDSTEP;
JUMP FIN, SEV, GT,1;
STEP LM2, ....;
JUMP CAS, SEV, EQ, 1;
ASSIGN IFN1, FILE1, ....;
JUMP ENDLM2;
CAS; ASSIGN IFN1, FILE2,....; ENDLM2: ENDSTEP;
FIN:
$ENDJOB;
```

Si l'activité LM1 donne un code de gravité d'erreur égal à 0, l'activité LM2 est exécutée et FILE1 est affecté à IFN1.

Si l'activité LM1 donne un code de gravité d'erreur égal à 1, l'activité LM2 est exécutée et FILE2 est affecté à IFN1.

Dans tous les autres cas, l'ordre JUMP qui suit l'ordre ENDSTEP de l'activité LM2 provoque un branchement sur l'ordre JCL portant l'étiquette FIN (autrement dit, sur \$ENDJOB).

**Exemple 2 :**

En cas d'arrêt prématuré d'un module chargeable, l'examen de la description de travail se poursuit séquentiellement jusqu'à la détection de l'un des ordres JCL suivants :

\$ENDJOB	Arrêt anormal du travail.
STEP <sup>*</sup>	Arrêt anormal du travail.
JUMP	Test des codes d'état (ou des inverseurs logiques). Si la condition est vérifiée, l'exécution du travail se poursuit normalement, après passage à l'ordre portant l'étiquette spécifiée.

\* Il s'agit d'un ordre JCL de base, mais le résultat serait le même pour un ordre JCL étendu dont le développement constituerait une description d'activité.

Considérons le test suivant :

```
$JOB FIRSTJOB,....;
STEP LM1, ...;
.
.
.
ENDSTEP;
STEP LM2, ...;
.
.
.
ENDSTEP;
JUMP CONTINUE;
RUN OTHERJOB, ...;
STEP LM3, ...;
.
.
.
ENDSTEP;
$ENDJOB;
```

L'activité LM2 n'est exécutée que si l'activité LM1 se termine normalement.

En revanche, le paramètre CONTINUE étant spécifié dans l'ordre JUMP, OTHERJOB est introduit et l'activité LM3 est exécutée quel que soit le résultat de l'activité LM2.

**Exemple 3 :**

```
$JOB SWITCHER, ...;
START: STEP LM1, ...;
.
.
.
ENDSTEP;
JUMP T3, SW1, EQ, 1;
LET SW1, 1;
STEP LM2, ...;
.
.
.
ENDSTEP;
```

```

JUMP START;
T3: STEP LM3, ....;
.
.
.
ENDSTEP;
$ENDJOB;

```

L'inverseur SW1 étant initialisé à 0 au moment du lancement du travail, l'ordre des opérations est le suivant :

- Exécution de l'activité LM1, puis de l'activité LM2 si SW1 est toujours à 0 ; dans le cas contraire, branchement sur l'activité LM3 (qui porte l'étiquette T3).
- Branchement inconditionnel vers l'amont sur l'activité LM1 (qui porte l'étiquette START).
- Exécution de l'activité LM3 (qui porte l'étiquette T3).

**Exemple 4 :**

```

$JOB .... ;
.
.
LIB SL INLIB1=LIBX;
JUMP STEP2, SW1, EQ, 1;
.
LIB SL INLIB1=LIBY;
STEP2: LIBMAINT SL LIB=LIBZ;
.
.
.
$ENDJOB;

```

Le but était d'utiliser comme bibliothèque d'entrée (pour l'activité LIBMAINT) .LIBX si SW1 était à 1, et .LIBY si SW1 était à 0.

LIB étant traité à la phase de traduction du travail alors que JUMP n'est traité qu'à l'exécution, cette formulation ne peut être prise en compte. A la traduction, les deux ordres LIB seront traités, le deuxième se substituant au premier. INLIB1=.LIBY sera donc le seul paramètre pris en compte pour LIBMAINT, quelle que soit la valeur de SW1 à l'exécution.

**Exemple 5 :**

```

$JOB .....;
.
.
.
VALUES X=FIRSTVAL;
.
.
.
X=FIRSTVAL, effectif
.
.
.
JUMP NEXT, SW1, EQ, 1;
MODVL X=SECONDVAL;
.
.
.
.
.
.
NEXT : STEP .....;
.
.
.
X=SECONDVAL, effectif
.
.
.
ENDSTEP;
.
.
.
$ENDJOB;

```

Les ordres VALUES et MODVL sont pris en compte à la traduction. MODVL attribue une nouvelle valeur à &X (SECONDVAL). Cette nouvelle valeur s'applique jusqu'à l'ordre \$ENDJOB (en supposant qu'il n'y ait pas d'autres ordres MODVL ou VALUES que les deux indiqués), qu'il y ait ou non un branchement sur NEXT à l'exécution. Donc, même si SW1 est à 1 et que les ordres figurant entre JUMP et NEXT:STEP sont sautés à l'exécution, &X conservera la valeur SECONDVAL entre MODVL et \$ENDJOB.

## 4.18 LET

### 4.18.1 Fonction

Positionne un inverseur système (ou tous les inverseurs système) ou force le code de gravité d'erreur d'une activité à une valeur entière de 0 à 4 inclus.

### 4.18.2 Rubrique

Description de travail ou d'activité.

### 4.18.3 Format

```
LET {SWi {0} {1}};  
     {SW hexa8 }  
     {SEV chiffre1 }
```

### 4.18.4 Description

L'ordre LET permet de forcer à 0 ou 1 un inverseur système devant être testé par un ordre JUMP postérieur faisant référence au même inverseur.

Il permet également de positionner en même temps, de gauche à droite, tous les inverseurs système, à une valeur hexadécimale donnée. L'utilisateur associe lui-même la condition à chaque inverseur logique.

L'ordre LET permet enfin d'affecter une valeur donnée au code de gravité d'erreur d'une activité, en dehors d'une rubrique de description d'activité.

#### 4.18.5 Paramètres obligatoires

##### **SWi**

Le paramètre SWi désigne l'un des 32 inverseurs système devant être forcés à 0 ou 1. "i" doit avoir une valeur comprise entre 0 et 31. Ces inverseurs logiques sont normalement initialisés à 0 au moment du lancement du travail. La valeur initiale d'un inverseur peut également être fixée soit par un ordre RUN ou RELEASE, soit par une commande opérateur MDJ ou EJR.

##### **SW**

Le paramètre SW désigne tous les inverseurs logiques système.

hexa8	Valeur de huit caractères hexadécimaux maximum, correspondant à 32 valeurs et permettant de positionner tous les inverseurs système à la fois (chaque valeur définissant, de gauche à droite, un inverseur de SW0 à SW31). Ainsi, FAB00000 forcera SW0, SW1, SW2, SW3, SW4, SW6, SW8, SW10 et SW11 à 1 et tous les autres inverseurs logiques à 0. (F (hexa)=1111 (binaire), A=1010 et B=1011).
-------	---

##### **SEV**

Le code de gravité d'erreur de l'activité est forcé à la valeur indiquée par "chiffre1", qui peut être un nombre entier de 0 à 4 inclus. Une fois l'exécution de l'ordre LET terminée, le travail se trouve dans le même état que si l'activité s'était terminée avec un code de gravité d'erreur de la valeur indiquée. Si cette valeur est 3 ou plus, le travail est donc arrêté prématurément ; cette situation peut être évitée par un ordre JUMP. Si plusieurs ordres LET SEV s'appliquent à la fois, seul le dernier est retenu.

Cette forme d'ordre LET est très utile en cas d'appel de séquence JCL par INVOKE. L'utilisateur peut, en insérant un ordre LET SEV à la fin d'une séquence appelée, forcer le code de gravité d'erreur à une valeur qui peut être testée dans le travail appelant, au moyen d'un ordre JUMP placé immédiatement après l'ordre INVOKE. Le résultat du test sera donc indépendant du code de gravité d'erreur résultant de la dernière activité exécutée dans la séquence d'ordres JCL appelée.

##### **Restriction**

La forme LET SEV ne doit pas figurer dans une rubrique de description d'entrées.

## 4.19 LIB

### 4.19.1 Fonction

Permet de définir un itinéraire de recherche dans les bibliothèques d'entrée, qui sera utilisé par un ordre JCL étendu ultérieur.

### 4.19.2 Rubrique

Description de travail.

### 4.19.3 Format

```

LIB {SL}
    {CU}
    {LM}  [ [ INLIB1= ] {TEMP
                           { (description-bibliothèque-d'entrée) } ]
    {SM}
    {BIN}

    [ [ INLIB2= ] {TEMP
                   { (description-bibliothèque-d'entrée) } ]
    [ [ INLIB3= ] {TEMP
                   { (description-bibliothèque-d'entrée) } ]
    [ [ INLIB4= ] {TEMP
                   { (description-bibliothèque-d'entrée) } ]
    [ [ INLIB5= ] {TEMP
                   { (description-bibliothèque-d'entrée) } ] ;

```

#### 4.19.4 Description

L'ordre LIB définit un itinéraire de recherche dans les bibliothèques d'entrée, qui sera utilisé dans un ordre JCL étendu ultérieur. En d'autres termes, la recherche de l'unité de bibliothèque concernée s'effectue en examinant d'abord INLIB1, puis INLIB2, puis INLIB3, etc. Si un même nom d'unité existe dans plusieurs bibliothèques, c'est la première unité rencontrée qui sera traitée. L'utilisateur peut également limiter la recherche à une bibliothèque spécifique en indiquant son numéro INLIB dans l'ordre JCL étendu.

L'ordre LIB peut apparaître n'importe où dans une rubrique de description de travail. Un ordre LIB demeure valable jusqu'à ce qu'il soit annulé par un nouvel ordre LIB du même type (c'est-à-dire SL, CU, LM, SM ou BIN). LIB sert principalement à définir des bibliothèques d'entrée pour un ordre LIBMAINT ultérieur. LIB SL peut être utilisé pour spécifier les entrées des compilateurs (COBOL, FORTRAN, etc.), et LIB CU pour indiquer les entrées de l'éditeur de liens (LINKER).

La longueur maximum de l'itinéraire de recherche défini par LIB dépend de type de bibliothèque concerné. Les longueurs maximum sont les suivantes :

- 3 pour SL, SM et BIN
- 1 pour LM
- 5 pour CU

Un itinéraire de recherche ne doit pas comporter de "trou" : autrement dit, si INLIB3 est spécifié, INLIB1 et INLIB2 doivent l'être également ; si, par exemple, l'itinéraire se compose de deux bibliothèques, il doit s'agir de INLIB1 et INLIB2, et non de INLIB1 et INLIB3 ou INLIB2 et INLIB3.

Les mots-clés "INLIBi=" peuvent être omis, auquel cas la première bibliothèque mentionnée est associée à INLIB1, la deuxième à INLIB2, etc.

Contrairement à l'ordre JCL "VALUES", un ordre LIB figurant dans une séquence JCL principale demeure valable dans une séquence appelée (par l'ordre JCL "INVOKE") jusqu'à l'apparition éventuelle d'un nouvel ordre LIB à l'intérieur de la séquence appelée. Un ordre LIB figurant dans une séquence appelée n'est valable qu'à l'intérieur de cette séquence. Ainsi, après l'ordre INVOKE, c'est le dernier ordre LIB de la séquence JCL principale qui est à nouveau pris en compte dans cette dernière.

Un ordre LIB figurant dans une séquence JCL principale n'est pas valable pour une séquence JCL exécutée (par l'ordre JCL EXECUTE). Un ordre LIB faisant partie d'une séquence JCL exécutée n'est valable qu'à l'intérieur de cette séquence (et ne peut donc être pris en compte dans la séquence principale après l'ordre EXECUTE).

Un ordre LIB figurant dans une séquence JCL principale est pris en compte dans une séquence JCL "insérée" dans la séquence principale au moyen de l'ordre \$SWINPUT. Contrairement à ce qui se passe pour les ordres INVOKE et EXECUTE, un ordre LIB appartenant à la séquence insérée (au moyen de l'ordre \$SWINPUT) s'applique à la séquence principale après l'ordre \$SWINPUT.

L'ordre LIB étant traité à la traduction, l'ordre JUMP (qui, lui, n'est pris en compte qu'à l'exécution du travail) ne peut être utilisé pour "sauter" un ordre LIB. Pour plus de précisions sur l'utilisation de LIB avec LIBMAINT, voir le *Guide utilisateur de maintenance des bibliothèques*.

## 4.19.5 Paramètres

#### 4.19.5.1 Paramètre obligatoire

## description-bibliothèque-d'entrée

Ce paramètre permet de définir une bibliothèque d'entrée à utiliser dans l'itinéraire de recherche. TEMP signifie qu'il s'agit d'une bibliothèque temporaire. Si le mot-clé "INLIBi=" est mentionné, "i" indique le rang de la bibliothèque dans l'itinéraire de recherche. Si "INLIBi=" est omis, c'est la position de la bibliothèque dans l'ordre LIB qui détermine son rang dans l'itinéraire de recherche.

#### 4.19.5.2 Paramètres facultatifs

Les paramètres facultatifs ci-dessous spécifient le type de bibliothèque :

SL	Bibliothèques origine. C'est la valeur implicite.
CU	Bibliothèques d'unités compilées.
LM	Bibliothèques de modules chargeables.
BIN	Bibliothèques binaires.

### **Exemples :**

```
$JOB . . . . . ;  
.  
.  
.  
1 LIB SL INLIB1 = (.MYLIB1);  
.  
.  
.  
.  
2 LIB SL INLIB1 = (.MYLIB2);  
.  
.  
.  
.  
3 LIB CU INLIB1=(.MYLIBX)  
    INLIB2=TEMP;  
.  
.  
.  
.  
$ENDJOB;
```

} champ d'application de 1

} champ d'application de 2

} champ d'application de 2 et 3

Le premier ordre LIB définit .MYLIBX comme bibliothèque SL d'entrée.  
Le deuxième ordre LIB annule le premier : la bibliothèque d'entrée devient MYLIBY.  
Le troisième ordre LIB définit un itinéraire de recherche dans les bibliothèques CU. Cet ordre LIB concernant un autre type de bibliothèque (CU en l'occurrence), il n'annule pas l'ordre LIB SL, et .MYLIBY reste donc valable.

Les ordres LIB étant traités à la traduction, ils ne sont pas affectés par la présence d'ordres JUMP (traités à l'exécution) dans le travail. Ainsi, quel que soit l'itinéraire d'exécution du travail (itinéraire déterminé par les ordres JUMP), chaque ordre LIB annule le précédent ; le champ d'application de chacun de ces ordres est indiqué ci-dessus.

## 4.20 MESSAGE

### 4.20.1 Fonction

Permet d'envoyer un message à l'opérateur de la console pendant la traduction du JCL.

### 4.20.2 Rubrique

Description de travail.

### 4.20.3 Format

```
MESSAGE 'chaîne105' ;
```

### 4.20.4 Description

L'ordre MESSAGE permet à l'utilisateur d'afficher un message sur la console opérateur au moment de la traduction du JCL.

La longueur du message ne doit pas dépasser 105 caractères.

## 4.21 MODVL

### 4.21.1 Fonction

Permet de modifier les valeurs implicites des paramètres dans les ordres JCL.

### 4.21.2 Rubrique

Description de travail, séquence INVOKE ou EXECUTE.

### 4.21.3 Format

```
MODVL [valeur-paramètre-1 [,valeur-paramètre-2]...]  
[mot-clé1=valeur-mot-clé-1 [mot-clé2=valeur-mot-clé-2]...] ;
```

### 4.21.4 Description

L'ordre MODVL permet de modifier les valeurs en cours des paramètres, les seuls affectés étant ceux explicitement spécifiés dans MODVL. Par ailleurs, l'ordre MODVL a le même effet que l'ordre VALUES. Un ordre MODVL n'annule pas l'effet d'un ordre MODVL ou VALUES précédent autrement que pour les paramètres dont les nouvelles valeurs sont spécifiées.

Les nouvelles valeurs spécifiées par un ordre MODVL s'appliquent à tous les ordres JCL suivants jusqu'à détection d'un autre ordre MODVL, d'un ordre VALUES, de la fin de la séquence JCL considérée ou de la fin de la rubrique de description du travail. Les spécifications de l'ordre MODVL peuvent porter sur des valeurs omises dans le paramètre VALUES des ordres INVOKE, EXECUTE et RUN ou de la commande opérateur EJR.

L'ordre MODVL ne permet pas de forcer des valeurs dans les ordres \$JOB, \$INPUT, \$ENDINPUT, \$DATA ou \$SWINPUT, ni dans un ordre VALUES suivant. "Valeur-paramètre-i" remplace la valeur du paramètre positionnel "&i" et "valeur-mot-clé-i" remplace la valeur du paramètre à mot-clé "mot-clé-i". Les paramètres positionnels et à mot-clé sont décrits au chapitre 2 et les règles concernant le remplacement des paramètres dans les ordres JCL sont présentées dans le *Guide utilisateur JCL*.

L'ordre MODVL a un effet cumulatif en ce sens qu'il donne une valeur à un paramètre qui n'en avait pas auparavant, ou donne des valeurs à de nouveaux paramètres non encore spécifiés, ou change l'ancienne valeur d'un paramètre existant, mais en laissant tels quels les paramètres qu'il ne spécifie pas explicitement.

L'ordre VALUES annule au contraire les paramètres qui ne sont pas explicitement spécifiés. Pour annuler la valeur d'un paramètre au moyen de l'ordre MODVL, il faut spécifier NIL.

Lorsque la virgule est utilisée comme séparateur, les paramètres positionnels omis sont signalés par autant de virgules, sans aucune valeur devant. Lorsque l'espace est utilisé comme séparateur, les paramètres omis sont signalés par autant de caractères "#". Il est inutile de signaler les paramètres omis à droite de la dernière valeur spécifiée par des virgules ou des caractères "#" (voir l'exemple 1 ci-après). Les paramètres à mot-clé omis ne sont pas signalés, étant identifiés par le mot-clé et non par leur position.

L'ordre MODVL modifie les valeurs de paramètres d'un ordre JCL de gauche à droite.

#### 4.21.5 Paramètres facultatifs

##### **valeur-paramètre-i**

Ce paramètre fixe la nouvelle valeur implicite de la chaîne &i (où  $1 < i < 99$ ) dans une séquence JCL appelée ou dans une rubrique de description de travail.

La nouvelle valeur spécifiée remplace l'ancienne valeur éventuellement donnée par un ordre VALUES ou MODVL précédent. La nouvelle valeur doit être conforme aux règles syntaxiques du ou des paramètres auxquels elle s'applique. Il peut s'agir d'une chaîne protégée (chaîne de caractères entre apostrophes, les apostrophes qui apparaissent à l'intérieur de la chaîne devant être doublées) de 128 caractères maximum. La position d'un paramètre par rapport à d'autres paramètres dans un ordre MODVL définit la valeur "i" pour l'identification de la valeur "i" associée.

##### **valeur-mot-clé-i**

Ce paramètre fixe la nouvelle valeur implicite devant remplacer chaque occurrence de "&mot-cléi" dans une séquence JCL appelée ou dans une rubrique de description de travail. La nouvelle valeur spécifiée remplace l'ancienne valeur éventuellement donnée par un ordre VALUES ou MODVL précédent. La nouvelle valeur spécifiée doit être conforme aux règles syntaxiques du ou des paramètres auxquels elle s'applique. Il peut s'agir d'une chaîne protégée de 128 caractères maximum. La longueur d'un mot-clé ne doit pas dépasser huit caractères.

***Exemple 1 :***

Il y a 10 paramètres positionnels (&1 à &10 compris) et il s'agit de donner les valeurs p2, p4, p6 aux paramètres 2, 4 et 6 respectivement (c'est-à-dire aux paramètres &2, &4 et &6). Les quatre ordres MODVL ci-dessous sont équivalents :

```
MODVL , p2,, p4,, p6,,,,;
MODVL , p2,, p4,, p6;
MODVL # p2 # p4 # p6 # # # #;
MODVL # p2 # p4 # p6;
```

Les anciennes valeurs éventuelles de &2, &4 et &6 sont remplacées par p2, p4 et p6 respectivement. Le fait que &2 n'ait pas eu de valeur ou ait eu une valeur spécifiée par un ordre VALUES ou MODVL précédent n'a aucune importance ; sa valeur est maintenant p2, et elle le restera jusqu'à modification par un autre ordre MODVL ou VALUES.

Les paramètres pour lesquels de nouvelles valeurs ne sont pas spécifiées ne sont pas affectés par l'ordre MODVL. Les paramètres &1, &3, &5, &7, &8, &9 et &10 gardent donc les éventuelles valeurs qu'ils avaient avant l'exécution de l'ordre MODVL.

***Exemple 2 :***

Dans la séquence suivante :

```
VALUES      DK1='MD=K116    DVC=MS/M500'
            DK2='MD=K200    DVC=MS/M500' ;
```

&DK1 spécifie MD=K116 DVC=MS/M500.  
&DK2 spécifie MD=K200 DVC=MS/M500.

Dans la séquence suivante :

```
MODVL      DK1='MD=C200    DVC=MS/M500' ;
```

&DK1 spécifie MD=C200 DVC=MS/M500.  
&DK2 spécifie MD=K200 DVC=MS/M500.

## 4.22 OUTVAL

### 4.22.1 Fonction

Permet de fixer la valeur implicite des paramètres de gestion et de mise en forme des sorties des ordres SYSOUT et WRITER dans le travail en cours.

### 4.22.2 Rubrique

Description de travail.

### 4.22.3 Format

```
OUTVAL [CLASS=identificateur1]
        [PRIORITY=chiffre1]
        [WHEN={JOB      }]
        [      {STEP     }]
        [      {IMMED   }]
        [      {DEFER   }]
        [      {chiffres5}]
        [ {HOLD  }]
        [ {NHOLD}]
        [NAME=identificateur8]
        [ {BANNER }]
        [ {NBANNER} ]
        [BANLEVEL={JOB      }]
        [      {OUTPUT  }]
        [      {COPY     }]
        [      {MEMBER  }]
        [BANINF=(alphanum12 [,alphanum12]...)]
        [COPIES=chiffres2]
        [DEVCLASS=classe-d'appareils [MEDIA=nom-volume]]
        [DEST=[nom-hôte] .nom-station]
        [ {SLEW  }]
        [ {NSLEW}]
        [ {DELETE }]
        [ {NDELETE}];
```

#### 4.22.4 Description

L'ordre OUTVAL modifie les valeurs implicites système applicables à tous les fichiers SYSOUT du travail en cours. Il ne peut figurer à l'intérieur d'une rubrique de description d'activité, mais peut être spécifié entre les activités. Il reste valable jusqu'au prochain ordre OUTVAL (ou, s'il n'y a plus d'ordre OUTVAL, jusqu'à la fin du travail).

La modification des valeurs implicites s'effectue comme suit :

1. A l'initialisation du travail, les paramètres de sortie implicites sont les paramètres système standard.
2. A chaque ordre OUTVAL, les valeurs qui s'y trouvent spécifiées remplacent les valeurs implicites précédentes comme nouvelles valeurs implicites. Si aucun paramètre n'est spécifié dans OUTVAL (OUTVAL;), les paramètres système standard remplacent les valeurs implicites précédentes.
3. Chaque fois qu'un ordre WRITER ou SYSOUT est rencontré, les valeurs implicites de l'ordre OUTVAL précédent, s'il en existe un, sont utilisées comme valeurs implicites. Toute valeur de paramètre explicitement spécifiée dans WRITER ou SYSOUT l'emporte néanmoins temporairement sur les valeurs implicites (voir ci-dessous).

#### JOR et JOBOUT

Le système définit implicitement les caractéristiques de sortie du fichier JOB-REP contenant le JOR :

```
CLASS=C, PRIORITY=3, WHEN=JOB, COPIES=1
```

L'utilisateur peut néanmoins changer ces valeurs dans le premier ordre OUTVAL précédant la première activité, sauf la valeur JOB du paramètre WHEN, qui reste toujours valable quoi qu'il spécifie. Si aucun ordre OUTVAL ne figure avant la première activité ou si le premier ordre OUTVAL avant l'activité ne contient aucun paramètre, ce sont les valeurs implicites système qui s'appliquent. Un ordre OUTVAL de ce type affecte tous les ordres SYSOUT et WRITER qui suivent jusqu'au prochain ordre OUTVAL éventuel.

Le groupe de fichiers SYSOUT standard ayant les mêmes caractéristiques que JOB-REP et contenant les sorties du compilateur, de l'éditeur de liens et de l'activité, est considéré pour les sorties comme un seul fichier, appelé JOBOUT. L'éventuel ordre OUTVAL s'appliquant à JOB-REP détermine donc également quels sont les fichiers SYSOUT standard qui seront considérés comme faisant partie de JOBOUT.

#### 4.22.5 Paramètres

Pour la description des paramètres de l'ordre OUTVAL, voir l'ordre SYSOUT.

La classe de sorties, la classe d'appareils et le nom de volume définis ailleurs que dans OUTVAL l'emportent sur ce qui a été défini dans OUTVAL. Si une nouvelle classe est spécifiée, sa priorité prévaut donc sur celle qui a été spécifiée dans l'ordre OUTVAL.

## 4.23 POOL

### 4.23.1 Fonction

Demande l'affectation d'un ou plusieurs appareils de la même classe à un pool de périphériques, pour la durée d'une activité.

### 4.23.2 Rubrique

Description d'activité.

### 4.23.3 Format

```
POOL { [ {1
          { [ {chiffres2} ] classe-d'appareils [MAX=chiffres2] }
          {
            nom-d'appareil
          }
        } ;
      }
```

### 4.23.4 Description

L'ordre POOL permet d'affecter un ou plusieurs appareils d'une classe donnée à un pool de périphériques. Ce pool est réservé à l'usage d'une activité pendant toute sa durée.

Il est possible de spécifier plusieurs ordres POOL dans une activité, ce qui a pour effet d'ajouter des appareils au pool de périphériques de la classe correspondante.

Un ordre ASSIGN peut demander des appareils faisant partie d'un pool de périphériques. Dans ce cas, un ou plusieurs ordres POOL désignant des appareils de mêmes caractéristiques doivent précéder l'ordre ASSIGN dans la rubrique de description d'activité (voir le paramètre POOL dans l'ordre ASSIGN).

L'ordre POOL sert, dans certains cas, à limiter le nombre d'appareils affectés à une activité.

Supposons par exemple qu'un utilisateur veuille traiter deux fichiers enregistrés sur des volumes différents. Il doit donc spécifier deux ordres ASSIGN dans la rubrique de description de l'activité. Or, du point de vue du système, les affectations d'appareils se cumulent ; en d'autres termes, l'affectation de deux fichiers enregistrés sur des volumes différents est considérée comme une demande d'affectation de deux appareils.

Si l'utilisateur veut traiter ces deux fichiers séquentiellement, c'est-à-dire s'il veut que le second volume soit monté sur le même appareil que le premier une fois le traitement de celui-ci terminé, sachant qu'un seul appareil suffirait, il ne peut pas, sans l'ordre POOL, l'indiquer au système.

Cette impossibilité pose un problème dans les configurations comportant peu de périphériques disponibles, problème d'autant plus grave qu'il y a davantage d'ordres ASSIGN demandant des volumes différents.

L'utilisateur peut résoudre le problème en indiquant, dans les deux ordres ASSIGN relatifs aux deux fichiers, qu'il veut utiliser un appareil de pool et désigner le fichier (et donc le volume) à traiter en premier. Ensuite, pendant l'exécution de l'activité, il lance un ordre d'ouverture du second fichier lorsque le traitement du premier est terminé. Cet ordre se traduit par une demande de montage du second volume sur le même appareil de pool encore réservé à l'activité en cours.

### Commentaires et restrictions

- Les attributs de la classe d'appareils spécifiée dans les ordres ASSIGN correspondant à l'ordre POOL doivent être identiques à ceux de la classe d'appareils spécifiée dans cet ordre POOL.
- Si le nombre voulu d'appareils d'une classe donnée n'est pas immédiatement disponible, l'activité doit attendre que l'ordre POOL puisse être satisfait.
- Une fois l'activité lancée, le nombre requis d'appareils lui est garanti. Si une autre activité a affecté un fichier qui se trouve dans un volume monté sur l'un des appareils du pool affecté à l'activité, cette autre activité peut cependant accéder au fichier, sous réserve des restrictions de partage des fichiers qui lui sont applicables.

## 4.23.5 Paramètres

Les périphériques appartenant à un pool sont désignés par leur classe d'appareils ou leur nom d'appareil.

### 4.23.5.1 Paramètres obligatoires

#### classe-d'appareils

Indique la classe à laquelle appartiennent les périphériques du pool ainsi que ses attributs (voir chapitre 3). Cette forme d'identification est préférable à la forme "nom-d'appareil".

#### nom-d'appareil

Le format de ce paramètre est détaillé au chapitre 3. Lorsqu'un nom d'appareil est fourni, aucun autre paramètre d'identification d'appareil n'est nécessaire, et seul l'appareil mentionné est affecté au pool de périphériques. Pour affecter plus d'un périphérique à un pool en utilisant les noms d'appareils, un ordre POOL est donc nécessaire pour chaque appareil, et tous ceux qui sont mentionnés feront bien partie du même pool.

## 4.23.5.2 Paramètres facultatifs

Le premier paramètre facultatif s'utilise lorsqu'il faut affecter un nombre minimum de périphériques d'une classe donnée à un pool. La valeur implicite est 1 et la valeur maximum 99. La valeur indiquée doit être immédiatement suivie d'un astérisque (\*). L'activité n'est pas lancée tant que le nombre minimum d'appareils n'est pas disponible et qu'il ne lui a pas été affecté.

**MAX**

Ce paramètre indique le nombre maximum d'appareils à affecter, s'ils sont disponibles.

**Remarque :** La différence entre le nombre maximum et le nombre minimum d'appareils représente une demande conditionnelle d'appareils.

***Exemples :***

POOL 2\*MT/T9/D1600 ;

Cet ordre demande l'affectation de deux dérouleurs de bande 9 pistes avec une densité d'enregistrement de 1600 bits par pouce.

POOL MS/M500 ;

Cet ordre demande l'affectation d'une unité de disques MSU0555.

POOL MT/T9/D1600, MAX=2 ;

L'activité est lancée si au moins une unité de bande 9 pistes est disponible. L'autre unité de bande 9 pistes est demandée sous condition.

POOL 0\*MS/M452, MAX=1 ;

Cet ordre correspond à une demande conditionnelle d'unité de disques MSU0452. Dans ce cas, l'activité peut être lancée même si aucune unité de disques MSU0452 n'est disponible.

POOL MS02 ;

Cet ordre demande l'affectation d'une unité de disques dont l'identificateur est MS02.

## 4.24 PREFIX

### 4.24.1 Fonction

Définit un préfixe à accoler à tous les noms de fichiers externes commençant par un point(.) .

### 4.24.2 Rubrique

Description de travail.

### 4.24.3 Format

```
PREFIX {nom-préfixe} {nom-projet} ;
```

### 4.24.4 Description

L'ordre PREFIX sert à modifier le préfixe implicite des noms de fichiers externes. Le nom de préfixe implicite est le nom du projet auquel le travail est associé, qui est spécifié soit dans le paramètre PROJECT de l'ordre \$JOB, soit dans le catalogue de site.

Tout ordre PREFIX figurant dans une rubrique de description de travail remplace le précédent.

L'utilisation des préfixes est expliquée dans le *Manuel de gestion des catalogues*.

#### Commentaires et restrictions

- PREFIX ne s'applique pas aux noms de fichiers externes sous forme de chaîne protégée (entre apostrophes).
- Le nouveau nom de fichier externe, composé du nom de préfixe suivi de l'ancien nom de fichier externe (qui commence par un point), doit être conforme aux règles d'écriture des noms de fichiers externes (voir chapitre 3).
- Le nom de préfixe peut être composé de plusieurs noms séparés par un point (voir chapitre 3), ce qui permet à l'utilisateur de demander le préfixage automatique de plusieurs niveaux de la hiérarchie de catalogues (voir l'exemple 2 ci-après).

#### 4.24.5 Paramètre facultatif

##### **nom-préfixe**

La valeur indiquée pour "nom-préfixe" sera accolée au début de tout nom de fichier externe commençant par un point et n'étant pas une suite protégée.

"nom-préfixe" ne doit pas dépasser 42 caractères. S'il n'est pas spécifié, le préfixe sera le nom du projet.

##### ***Exemple 1 :***

```
PREFIX SALES;  
STEP.....;  
ASSIGN UPD, .ACCTS;  
ENDSTEP;
```

Dans cette séquence d'ordres, PREFIX remplace .ACCTS par SALES.ACCTS.

##### ***Exemple 2 :***

```
PREFIX AREA1.CUSTOMER;  
STEP ST1,....;  
ASSIGN IFN1, .ACCTS;  
ASSIGN IFN2, .ADRESSES;  
ENDSTEP;
```

Les deux ordres ASSIGN deviennent :

```
ASSIGN IFN1, AREA1.CUSTOMER.ACCTS;  
ASSIGN IFN2, AREA1.CUSTOMER.ADDRESSES;
```

## 4.25 QASSIGN

### 4.25.1 Fonction

Etablit la correspondance entre le nom symbolique et le nom externe d'une file d'attente, indique sa structure, sa méthode d'examen et le format des messages. Affecte également une file d'attente MCS à l'activité.

### 4.25.2 Rubrique

Description d'activité.

### 4.25.3 Format

```
QASSIGN
nom-file-d'attente-symbolique[.nom-sous-file-d'attente-symbolique1
                                [.nom-sous-file-d'attente-symbolique2
                                 [.nom-sous-file-d'attente-symbolique3]]]
nom-file-d'attente-externe1
[SITE=nom-noeud]
[ { IN
  [ { INOUT
        [ { ACCESS={ LIN } } ]
        [ { RR } } ]
  [ { OUT    [ REPLY=nom-file-d'attente-externe2 ] } ];
  }
```

### 4.25.4 Description

L'ordre QASSIGN sert à établir la correspondance entre le nom symbolique (interne) d'une file d'attente, indiqué dans la zone CD (Communication Description) du programme COBOL MCS (ou dans H\_CDOUT ou H\_CDIN d'une application GPL) et son nom externe, défini dans la description du réseau.

Il faut spécifier un ordre QASSIGN par file d'entrée utilisée dans une activité. Une ou plusieurs sous-fichiers d'attente peuvent être associées à un nom symbolique de file d'entrée. Chaque sous-fichier d'attente doit être identifiée par un ordre QASSIGN indiquant un nom symbolique de file d'attente et un nom symbolique de sous-fichier d'attente.

Une file de sortie symbolique ne nécessite pas obligatoirement un ordre QASSIGN (un nom symbolique de file de sortie peut être implicitement le nom du terminal correspondant).

Une rubrique de description d'activité peut comporter jusqu'à 26 ordres QASSIGN. Le premier ordre QASSIGN relatif à une file d'attente donnée définit la méthode d'examen des sous-fichiers d'attente correspondantes ; s'il est spécifié dans les ordres QASSIGN suivants relatifs à la même file d'attente, le paramètre ACCESS n'est alors pas pris en compte. Un nom de sous-fichier d'attente symbolique doit être unique à l'intérieur d'une rubrique de description d'activité.

Dans le cas des fichiers d'attente disque, il est inutile de spécifier un ordre ASSIGN, car l'ouverture du fichier s'effectue au niveau système et est donc partageable par tous les groupes de process.

Le mode de connexion entre les correspondants, ainsi que le format des messages en entrée et en sortie, peuvent être spécifiés. L'insertion automatique de caractères de fin de ligne (retour chariot et simple interligne) peut également être demandé.

Pour plus de précisions sur l'utilisation de l'ordre QASSIGN, voir le *Guide utilisateur MCS* et la série de manuels relative au *Système de gestion de communications CPF*.

## 4.25.5 Paramètres

### 4.25.5.1 Paramètres obligatoires

#### **nom-file-d'attente-symbolique**

Ce paramètre indique le nom de la file d'attente comme suit :

- en COBOL MCS, dans "nom•donnée•1" de la rubrique CD (entrée et sortie) ;
- en GPL, dans QUEUE\_NAME de H\_CDOUT ou H\_CDIN.

"nom-file-d'attente-symbolique" ne doit pas dépasser 12 caractères (lettres, chiffres, trait d'union et caractère de soulignement).

#### **nom-file-d'attente-externe1**

Ce paramètre indique le nom de la file d'attente tel qu'il est déclaré à la génération du réseau. Ce nom ne doit pas dépasser 12 caractères (lettres, chiffres, trait d'union et caractère de soulignement).

## 4.25.5.2 Paramètres facultatifs

**nom-sous-file-d'attente-symbolique**

Ce paramètre identifie une sous-fille d'attente associée à la file d'attente désignée. Il ne doit pas dépasser 12 caractères (lettres, chiffres, trait d'union et caractère de soulignement). Il n'est pas pris en compte si OUT ou INOUT est spécifié, et s'applique uniquement aux files d'entrées, qu'il fractionne logiquement comme suit :

- en COBOL MCS, "nom•donnée•2", "nom•donnée•3" et "nom•donnée•4" de la rubrique CD d'entrée ;
- en GPL, SUBQUEUE\_NAME, SUBQUEUE2\_NAME et SUBQUEUE3\_NAME de H\_CDIN.

**IN**

Ce paramètre indique que la file d'attente spécifiée est une file d'entrée. Il s'applique uniquement aux files d'attente de programmes. C'est la valeur implicite.

**INOUT**

Traitement en entrée et en sortie à la fois. Une sous-fille d'attente ne peut pas être spécifiée en mode INOUT. Ce paramètre s'applique uniquement aux files d'attente de programmes.

**OUT**

Ce paramètre indique que la file d'attente spécifiée est une file de sortie. Une sous-fille d'attente ne peut pas être spécifiée en mode OUT.

**ACCESS**

Ce paramètre spécifie la méthode d'examen des sous-filles d'attente en réception. ACCESS n'est pas pris en compte lorsque OUT est spécifié.

LIN	Chaque réception lance l'examen des sous-filles d'attente, en commençant par la première. C'est la valeur implicite.
RR	Chaque réception lance l'examen des sous-filles d'attente, à partir de celle qui suit la dernière sous-fille ayant fourni un message lors de la précédente réception.

## **REPLY**

Ce paramètre spécifie le nom externe de la file d'attente où sont envoyés les messages émis pendant l'activité en cours (n'est applicable qu'aux paramètres INOUT et OUT).

En COBOL, ce nom doit figurer dans la zone SYMBOLIC SOURCE de la zone CD d'entrée du destinataire (voir la COMMUNICATION SECTION dans le *manuel de référence COBOL*).

En GPL, il doit être défini dans H\_CDIN.de manoeuvre avec labels standar  
REPLY s'applique uniquement aux communications inter-applications.

## **SITE**

Ce paramètre identifie un système DSA qui, pour le DPS 7000, peut être local (lsys ou distant (rsys).

**Remarque :** La longueur de ligne et le nombre de blocs sont définis à la génération du réseau. Le nombre de blocs correspond au nombre de lignes logiques à insérer dans un message envoyé par l'application à la file d'attente de terminal. La longueur de ligne spécifie la longueur de ligne logique.

## 4.26 RELEASE

### 4.26.1 Fonction

Libère un travail bloqué (à l'état HOLD) et le rend disponible pour la planification.

### 4.26.2 Rubrique

Description de travail.

### 4.26.3 Format

```
RELEASE nom-travail [{SWITCHES={hexa8}
[ { PASS } ]
[ { } ]
[ {SWITCHi={0} SWITCHj={0} }... ]
[ {1} {1} {1} ] }];
```

### 4.26.4 Description

L'ordre RELEASE est interprété au moment de l'exécution du travail comme la libération d'un travail bloqué (à l'état HOLD). Si deux travaux ou plus portent le nom de travail spécifié et ont les attributs correspondants (voir la restriction ci-après), ils sont tous libérés lors de l'exécution de RELEASE.

Lorsqu'un travail bloqué est associé à un compteur, l'ordre RELEASE décrémente le compteur de 1. Le travail bloqué n'est effectivement libéré que lorsque ce compteur est à zéro.

Les codes de gravité d'erreur résultant de l'exécution de RELEASE sont les suivants :

Codes	Signification
0	RELEASE correctement exécuté.
1	Le travail ou les travaux sur lesquels porte l'ordre RELEASE ne sont pas à l'état HOLD. L'ordre n'est donc pas pris en compte.
2	Aucun travail du système n'a le nom de travail indiqué et les attributs correspondants (voir restriction ci-après).

#### Restriction

Le travail à libérer doit avoir les mêmes attributs (nom d'utilisateur et nom de projet) que le travail contenant l'ordre RELEASE.

## 4.26.5 Paramètres

### 4.26.5.1 Paramètre obligatoire

#### **Nom-travail**

Ce paramètre identifie le travail à libérer. Sa description complète figure dans l'ordre \$JOB.

### 4.26.5.2 Paramètres facultatifs

#### **SWITCHES**

Ce paramètre indique la valeur initiale de tous les inverseurs système dans le travail libéré.

hexa8	Valeur de huit caractères hexadécimaux maximum, indiquant la valeur binaire des différents inverseurs logiques. Lorsque moins de huit caractères sont spécifiés, l'espace restant à droite est garni de zéros.
PASS	Cette valeur indique que les inverseurs logiques du travail libéré doivent être positionnés à la même valeur que ceux du travail à l'origine de cette libération au moment de l'exécution de RELEASE.

#### **SWITCHi**

Les inverseurs spécifiés sont positionnés à la valeur voulue. Les autres inverseurs restent à la valeur qu'ils avaient dans le travail bloqué avant l'exécution de l'ordre RELEASE.

## 4.27 REPORT

### 4.27.1 Fonction

Consigne un message dans le JOR au moment de l'exécution.

### 4.27.2 Rubrique

Description de travail ou d'activité.

### 4.27.3 Format

```
REPORT 'chaîne110';
```

### 4.27.4 Description

L'ordre REPORT permet à l'utilisateur de faire apparaître un message dans la partie exécution du JOR au moment de l'exécution. Ce message ne doit pas dépasser 110 caractères.

REPORT doit être utilisé de préférence à SEND dans toute la mesure du possible, afin de ne pas faire appel à l'opérateur sans nécessité absolue.

L'ordre REPORT peut être utilisé avec l'ordre JUMP pour obtenir, dans le JOR, le suivi d'exécution d'une description de travail donnée, ou pour résumer le déroulement de l'exécution d'un travail.

## 4.28 RUN

### 4.28.1 Fonction

Demande l'exécution d'un travail indépendant du travail qui émet l'ordre RUN.

### 4.28.2 Rubrique

Description de travail.

### 4.28.3 Format

```

RUN
  {[ INFILE=] (description-fichier-d'entrée-séquentiel)
  { [INDEF = (paramètres-define)] }
  { {nom-unité-bibliothèque}
  { { } [INLIB=](description-bibliothèque-d'entrée) }
  { {nom-avec-astérisque} }
  [ INFILE2=(description-fichier-d'entrée-séquentiel)
    [INDEF2=(paramètres-define)] ]
  [ INFILE3=(description-fichier-d'entrée-séquentiel)
    [INDEF3=(paramètres-define)] ]
  [ INFILE4=(description-fichier-d'entrée-séquentiel)
    [INDEF4=(paramètres-define)] ]
  [CLASS=identificateur2]
  [HOLD]
  [HOLDOUT]
  [PRIORITY=chiffre1]
  [JOBS=([nom-travail1] [nom-travail2])]
  [VALUES=([valeur-paramètre1 [valeur-paramètre2]...]
    [mot-clé1=valeur-mot-clé1 [mot-clé2=valeur-mot-clé2]...])]
  [ {SWITCHES={hexa8}          }]
  [ {          {PASS }        }]
  [ {          { }          }]
  [ {SWITCHi={0} SWITCHj={0}    ]...    }]
  [ {          {1}          {1} }        ]
  [DELETE]
  [SIZEOPT=(paramètres-size) ]
  [STEPOPT=(paramètres-step) ];

```

#### 4.28.4 Description

L'ordre RUN demande l'introduction dans le système d'un train de travaux distinct de celui dans lequel il figure (ce train de travaux est dit "généré dynamiquement"). L'ordre JCL RUN et la commande opérateur EJR ont la même fonction. L'unité de bibliothèque sur disque ou le fichier d'entrée séquentiel spécifié dans l'ordre RUN doit avoir la structure d'un train de travaux d'entrée.

L'effet de RUN est le même que si l'opérateur avait lancé le lecteur des entrées pour lire le train de travaux spécifié.

Les données peuvent être en format DATA ou DATASSF. Les fichiers non natifs peuvent être utilisés si leurs caractéristiques sont définies par le groupe de paramètres INDEF. Dans tous les cas, les caractéristiques des fichiers doivent être compatibles avec les règles du lecteur des entrées (par exemple, la longueur d'article ne doit pas dépasser 255 caractères). Une entrée secondaire peut provenir de fichiers définis par INFILE2, INFILE3 et INFILE4 si un ordre \$SWINPUT fait référence à ces fichiers.

Le paramètre JOBS permet à l'utilisateur de sélectionner un travail ou une suite de travaux du train contenu dans les fichiers spécifiés. Si JOBS est omis, tous les travaux du train sont sélectionnés pour constituer le train de travaux d'entrée. Il faut noter que si des droits d'accès ont été déterminés, seul un projet opérateur peut exécuter un travail dont le nom d'utilisateur diffère de celui du travail appelant.

Si CLASS, PRIORITY, HOLD ou HOLDOUT sont spécifiés dans RUN, leurs valeurs remplacent celles des ordres \$JOB relatifs aux travaux sélectionnés.

Pour la soumission d'un seul travail, les ordres \$JOB et \$ENDJOB ne sont pas nécessaires. Dans ce cas, les valeurs implicites suivantes s'appliquent : le nom de travail est le nom d'unité de bibliothèque (RUN nom-unité-bibliothèque) ou le nom de fichier externe (RUN nom-fichier-externe). Lorsque ce nom dépasse huit caractères, seuls les huit premiers sont retenus.

Si VALUES est spécifié dans RUN, les travaux générés dynamiquement sont mis à jour (valeurs des paramètres positionnels ou des mots-clés).

Si une relance à chaud est exécutée après un incident survenu pendant l'exécution de l'activité RUN (RUN génère sa propre activité), le traitement du train d'entrées reprend à partir du travail qui suit le dernier travail normalement introduit dans le système.

**Remarque :** L'ordre RUN définit une activité complète, mais est considéré comme faisant partie du système d'exploitation de base et est donc traité dans le présent manuel comme un ordre JCL de base.

## 4.28.5 Paramètres

### 4.28.5.1 Paramètres obligatoires

Le seul groupe de paramètres obligatoires est celui identifiant le fichier qui contient le train de travaux. Ce fichier peut être spécifié comme un fichier d'entrée séquentiel (éventuellement précédé de "INFILE=") ou comme une (ou plusieurs) unité(s) d'une bibliothèque d'entrée (éventuellement précédée(s) de "INLIB="). Les paramètres de la description d'un fichier d'entrée séquentiel et de la description d'une bibliothèque d'entrée sont donnés au chapitre 3.

Dans le cas d'une bibliothèque d'entrée, l'unité de bibliothèque (c'est-à-dire le sous-fichier de bibliothèque) peut être identifiée :

- soit par son nom d'unité, qui ne doit pas dépasser 31 caractères (lettres, chiffres, trait d'union, caractère de soulignement), et doit commencer par une lettre ou un chiffre ;
- soit par un nom utilisant la convention astérisque (voir le manuel de référence relatif à la maintenance des bibliothèques - 01UP). L'astérisque représentant plusieurs caractères, son utilisation permet de sélectionner une ou plusieurs unités d'une bibliothèque, la sélection de toutes les unités d'une bibliothèque étant un cas particulier. Les unités sélectionnées sont incorporées dans l'ordre de leurs noms (ordre de classement EBCDIC).

La ou les descriptions de travail doivent être au complet dans chaque unité de bibliothèque. INLIB=SYS identifie la bibliothèque système SYS.HSLLIB.

### 4.28.5.2 Paramètres facultatifs

#### **INDEF**

Ce paramètre permet d'associer les paramètres de définition (voir chapitre 3) au fichier d'entrée séquentiel contenant le train de travaux.

#### **INFILEi**

Ces paramètres spécifient les fichiers d'entrée secondaires ( $2 < i < 4$ ) cités dans des ordres \$SWINPUT. Le contenu de ces fichiers est incorporé dans le train de travaux à l'adresse des ordres \$SWINPUT correspondants. La présence de ces paramètres dans l'ordre RUN garantit la disponibilité des ressources au moment de l'exécution.

#### **INDEFi**

Ces paramètres spécifient les paramètres de définition à associer aux fichiers désignés dans les paramètres INFILEi correspondants ( $2 < i < 4$ ).

## **CLASS**

Ce paramètre indique la classe à laquelle appartiennent les travaux générés dynamiquement (voir l'ordre \$JOB pour plus de détails).

La classe indiquée dans ce paramètre remplace celle indiquée dans les ordres \$JOB du train de travaux généré dynamiquement.

## **HOLD**

Ce paramètre bloque les travaux sélectionnés (voir le paramètre HOLD de l'ordre \$JOB).

## **HOLDOUT**

Ce paramètre bloque les fichiers de sortie générés par les travaux sélectionnés jusqu'à ce que l'opérateur les libère au moyen d'une commande RO.

## **PRIORITY**

Ce paramètre indique une priorité de planification d'un chiffre, comprise entre 0 (priorité la plus élevée) et 7 (priorité la plus basse). Les travaux sélectionnés sont planifiés en fonction de cette priorité, qui prévaut sur celle éventuellement spécifiée dans leur ordre \$JOB. Dans le cadre d'une priorité donnée, les travaux sont planifiés suivant la règle du premier entré-premier sorti (FIFO).

**Remarque :** Un travail, même de priorité élevée, n'est pas planifié si le nombre maximum de travaux de la même classe a déjà été planifié.

## **JOBS**

Ce paramètre permet à l'utilisateur de sélectionner une suite de travaux dans le train de travaux enregistré. "nom-travail1" est le nom du premier travail de la séquence et "nom-travail2" celui du dernier. Ainsi, tous les travaux de "nom-travail1" à "nom-travail2" sont sélectionnés dans le fichier d'entrée. JOBS ne doit pas être spécifié lorsque plusieurs unités de bibliothèque ont été désignées en utilisant la convention astérisque.

Si "nom-travail1" n'est pas spécifié, la séquence commence par le premier travail du fichier d'entrée. Si "nom-travail2" n'est pas spécifié, la séquence se termine par le dernier travail du fichier d'entrée.

Si le paramètre JOBS n'est pas spécifié, tous les travaux que contient le fichier d'entrée sont sélectionnés pour constituer le train de travaux en entrée.

**Remarque :** "JOBS=(JBA,JBA)" peut s'écrire "JOBS=JBA". Cet ordre signifie qu'un seul travail, JBA, sera sélectionné.

En résumé :

JOBS non spécifié	Sélection de tous les travaux
JOBS=X ou JOBS=(X,X) ou JOBS=(X X)	Sélection du travail X
JOBS=(,X)	Sélection de tous les travaux jusqu'au travail X inclus
JOBS=(X,)	Sélection de tous les travaux à partir du travail X inclus
JOBS=(X,Y) ou JOBS=(X Y)	Sélection de tous les travaux du travail X au travail Y inclus

### **VALUES**

Ce paramètre spécifie les valeurs de paramètres qui seront transmises à la séquence JCL. Voir l'ordre INVOKE pour la description de VALUES, et le *Guide utilisateur JCL* pour le remplacement de paramètres.

### **SWITCHES**

Ce paramètre spécifie la valeur initiale de tous les inverseurs système dans les travaux sélectionnés.

hexa8	Valeur de huit caractères hexadécimaux indiquant la valeur binaire des différents inverseurs logiques, de gauche à droite. Exemple : SWITCHES=40000001 positionne les inverseurs SW1 et SW31 à 1. Lorsque moins de huit caractères sont spécifiés, l'espace restant à droite est garni de zéros.
PASS	Si cette valeur est spécifiée, les inverseurs logiques de tous les travaux sélectionnés seront forcés aux mêmes valeurs que ceux du travail appelant (travail contenant l'ordre RUN) au moment de l'exécution de l'activité RUN.

### **SWITCHi**

Les inverseurs logiques spécifiés sont forcés aux valeurs indiquées. Les inverseurs non spécifiés ont une valeur initiale de 0 (au contraire de SWITCHi dans l'ordre RELEASE).

### **DELETE**

Ce paramètre est pris en compte uniquement lorsque l'entrée provient d'une bibliothèque origine. Il permet la suppression d'une unité de bibliothèque après traitement par le lecteur des entrées. Si un nom avec astérisque a été utilisé pour la sélection des unités de bibliothèque, ces unités sont supprimées après traitement de la dernière d'entre elles par le lecteur des entrées. Lorsque le traitement d'une unité de bibliothèque ne peut pas être correctement effectué (par exemple, à la suite d'une erreur permanente d'E/S), l'unité n'est pas supprimée.

**SIZEOPT**

Permet de spécifier les paramètres de taille (voir le chapitre 3 et l'ordre SIZE).

**STEPOPT**

Ce paramètre permet à l'utilisateur de spécifier des paramètres commandant l'exécution de l'ordre RUN. Tous les paramètres de STEP, sauf DUMP=ALL, DEBUG et OPTIONS, peuvent être spécifiés.

***Exemple 1 :***

```
RUN DMJOB16, SYS;
```

Cet ordre introduit le train de travaux contenu dans l'unité DMJOB16 de la bibliothèque système SYS.HSLLIB.

***Exemple 2 :***

Si l'ordre RUN est le suivant :

```
RUN WJCIII, JSTREAM.TEST1, VALUES=(NEW.TFILE, FLSTAT=NIL),  
CLASS=D;
```

et si l'un des travaux demandés contient les ordres suivants :

```
$JOB JOB1, USER=BIG, PROJECT=YIN;  
.  
.  
.  
ASSIGN TFILE, &1, &FLSTAT;
```

le résultat est le suivant :

```
$JOB JOB1, USER=BIG, PROJECT=YIN, CLASS=D;  
.  
.  
.  
ASSIGN TFILE, NEW.TFILE;  
.  
.  
.
```

***Exemple 3 :***

```
RUN WELL1E, MST.STRS, JOBS=(,JBX3);
```

Cet ordre sélectionnera, dans le train de travaux que contient l'unité WELL1E de la bibliothèque MST.STRS, une suite de travaux allant du premier à JBX3.

***Exemple 4 :***

```
RUN QDLY, ACCT.PROC, JOBS=(NAT1, ), VALUES=(TAPE=X393);
```

Les travaux allant de NAT1 au dernier travail du train de travaux indiqué seront introduits dans le système, une fois toutes les occurrences de &TAPE remplacées par la chaîne X393 dans ces travaux.

## 4.29 SEND

### 4.29.1 Fonction

Envoie un message de l'utilisateur à l'opérateur.

### 4.29.2 Rubrique

Description de travail ou d'activité.

### 4.29.3 Format

```
SEND 'chaîne105'
[ {nom-utilisateur} ]
[ {MAIN} ] ;
```

### 4.29.4 Description

L'ordre SEND sert normalement à informer l'opérateur de certains besoins du traitement (par exemple, demande de davantage de bandes de manoeuvre pour le tri), ou s'utilise en fonction des résultats d'exécution d'un ordre JCL conditionnel (JUMP, LET).

Si l'utilisateur veut simplement signaler les résultats de l'exécution d'un ordre JCL sans que l'opérateur ait à intervenir, il doit utiliser l'ordre REPORT et non l'ordre SEND.

L'opérateur concerné est celui de la console associée au travail émetteur de l'ordre SEND, sauf si un nom d'utilisateur est spécifié ou s'il existe un ordre CONSOLE précédent dans la description du travail.

A noter que SEND prend effet à l'exécution du travail, alors que MESSAGE prend effet à la traduction du JCL.

## 4.29.5 Paramètres

### 4.29.5.1 Paramètre obligatoire

#### **'chaîne105'**

Le message est une suite de 105 caractères maximum, qui s'affichent à la console opérateur. L'interpréteur d'ordres ajoute le numéro de passage avant le message, au moment où celui-ci est envoyé à l'opérateur.

### 4.29.5.2 Paramètres facultatifs

#### **nom-utilisateur**

Identifie un utilisateur (et par là même, un terminal) en communication avec le système. Si ce paramètre est omis, le message apparaît implicitement à la console opérateur associée au travail émetteur de l'ordre SEND. Si un ordre CONSOLE figure précédemment dans la description du travail, le message apparaît à la console identifiée par cet ordre. Le paramètre "nom-utilisateur" permet à l'utilisateur de remplacer ces deux valeurs.

#### **MAIN**

Ce paramètre permet à l'utilisateur de remplacer un ordre CONSOLE antérieur pour demander que le message en cours soit envoyé à la console de l'opérateur principal (console système).

## 4.30 SIZE

### 4.30.1 Fonction

Définit la longueur de l'espace d'exécution nécessaire à l'exécution en mémoire d'une activité donnée.

### 4.30.2 Rubrique

Description d'activité.

### 4.30.3 Format

```
SIZE      [espace-d'exécution-déclaré]
          [CHPPAGE=chiffres]
          [NBBUF=chiffres4]
          [POOLSIZE=chiffres];
```

### 4.30.4 Description

L'ordre SIZE sert à déclarer la taille de l'espace d'exécution, en nombre de kilo-octets (1 Ko=1024 bits).

L'espace d'exécution déclaré est la quantité de mémoire nécessaire aux segments de code, aux segments de données et aux structures de gestion de process. Sont inclus dans ces segments les tampons de gestion de données utilisés pour les opérations d'E/S, qui sont déclarés à l'ouverture du fichier. L'ordre SIZE porte donc sur les segments résidant en mémoire de façon permanente pendant l'exécution (structures de gestion de process) et sur les segments chargés à partir de la mémoire de réserve au fur et à mesure des besoins pendant l'exécution (segments de code et segments de données).

L'exécution de l'activité concernée peut commencer seulement lorsque le système est en mesure d'offrir l'espace mémoire correspondant à la valeur du paramètre "espace-d'exécution-déclaré", ainsi que les autres ressources nécessaires (appareils, volumes, fichiers, etc.). Si le système ne peut satisfaire les besoins en mémoire au moment du lancement de l'activité, celle-ci attend cette ressource de la même façon qu'elle attendrait la libération d'un périphérique ou d'un volume (voir le *Guide utilisateur JCL*). Une fois l'activité en cours d'exécution, la place qui lui est disponible en mémoire peut différer de l'espace d'exécution initialement déclaré, suivant la charge du système et les besoins de l'activité en exécution, sauf si les options MAXMEM ou MINMEM de l'ordre STEP sont utilisées (voir l'ordre STEP). L'estimation initiale des besoins en mémoire d'une activité donnée peut s'effectuer au moyen de tables de valeurs (voir le *Manuel de l'administrateur système*).

Pour optimiser la valeur de l'espace d'exécution déclaré, il est possible d'utiliser le listage de l'éditeur de liens afin de connaître les valeurs correspondant aux structures de gestion de process, ainsi que le nombre et la taille des segments de code et de données. Le JOR fournit des informations sur les besoins en mémoire des tampons. Lorsque l'espace d'exécution déclaré ne peut pas être estimé de façon satisfaisante, il est possible d'utiliser l'option MAXMEM de l'ordre STEP avec l'ordre SIZE pour générer un graphe des performances (inversement proportionnelles au nombre de segments manquants indiqués dans le JOR) en fonction de la valeur de l'espace d'exécution déclaré. L'option MAXMEM indique au système que l'activité en cours d'exécution fait l'objet d'un test de performance et n'est jamais autorisée, quelle que soit la charge, à occuper plus de mémoire que n'en spécifie l'ordre SIZE. Si aucune valeur n'est déclarée, ou si la valeur d'exécution déclarée est inférieure à 200 Ko, l'espace d'exécution déclaré a une valeur de 200 Ko.

La longueur de page implicite du programme canal peut être remplacée dans les cas où celui-ci nécessite plus des quatre pages qui lui sont disponibles.

La dimension implicite du pool de tampons UFAS peut être remplacée au moyen du paramètre POOLSIZE de l'ordre SIZE. Cette dimension peut en effet ne pas convenir lorsqu'une activité accède à plusieurs fichiers (voir le *Guide utilisateur UFAS-EXTENDED*).

#### 4.30.5 Paramètres facultatifs

##### **espace-d'exécution-déclaré**

Spécifie l'espace mémoire nécessaire, en nombre de Ko. Il inclut la valeur explicite ou implicite de POOLSIZE. La valeur implicite de l'espace d'exécution déclaré est 200 Ko.

##### **CHPPAGE**

Ce paramètre spécifie, en octets, la longueur d'une page du programme canal. La valeur implicite est de 4 096 octets (4 Ko).

**NBBUF**

Ce paramètre indique le nombre de tampons nécessaires pour la mémoire verrouillée (structures de gestion de programme). NBBUF ne doit pas dépasser 4.000, et a une valeur implicite de 50.

**POOLSIZE**

Ce paramètre indique, en nombre de Ko, la taille maximum du pool de tampons UFAS. Cette taille est incluse dans la valeur (explicite ou implicite) de l'espace d'exécution déclaré ; elle ne doit pas lui être supérieure. La valeur implicite de POOLSIZE est inférieure de 15 Ko à celle de SIZE.

## 4.31 STEP

### 4.31.1 Fonction

Ouvre une rubrique de description d'activité et désigne un module chargeable à exécuter.

### 4.31.2 Rubrique

Description de travail.

### 4.31.3 Format

```

STEP
nom-module-chargeable (description-bibliothèque-d'entrée)

[XPRTY=chiffre1]

[CPTIME={9999999    }]
[      {chiffres7}]

[ELAPTIME={9999      }]
[      {chiffres4}]

[LINES={99999999    }]
[      {chiffres8}]

[DUMP={NO          }]
[      {DATA      [PRIVATE]}]
[      {ALL       [PRIVATE]}]

[DEBUG[=(description-fichier-d'entrée-séquentiel)]]

[OPTIONS='chaîne4096']

[REPEAT]

[ {MAXMEM} ]
[ {MINMEM} ];

```

#### 4.31.4 Description

L'ordre STEP est obligatoirement le premier ordre d'une rubrique de description d'activité. Il sert à demander le chargement du module chargeable dont l'exécution constitue l'activité. Le module chargeable est une unité d'une bibliothèque de modules chargeables, bibliothèque temporaire créée par le système ou bibliothèque utilisateur permanente, cataloguée ou non.

Les ordres qui suivent l'ordre STEP (et constituent la rubrique de description d'activité), demandent les ressources nécessaires à l'exécution de l'activité. A l'ordre STEP doit correspondre un ordre ENDSTEP, qui est obligatoirement le dernier ordre de la rubrique de description d'activité.

#### 4.31.5 Paramètres

##### 4.31.5.1 Paramètres obligatoires

###### **nom-module-chargeable**

Indique le module chargeable nécessaire pour l'exécution de l'activité. Ce nom ne doit pas dépasser 31 caractères (lettres, chiffres, trait d'union et caractère de soulignement) et doit commencer par une lettre ou un chiffre. C'est le premier paramètre positionnel de STEP.

###### **description-bibliothèque-d'entrée**

Identifie la bibliothèque où se trouve le module chargeable spécifié. La forme développée de ce paramètre figure au chapitre 3. C'est le second paramètre positionnel de STEP.

S'il s'agit de la bibliothèque temporaire standard TEMP.LMLIB, le paramètre TEMP est obligatoire.

SYS indique la bibliothèque système SYS.HMLIB.

## 4.31.5.2 Paramètres facultatifs

**XPRTY**

Ce paramètre indique une priorité d'exécution d'activité comprise entre 0 (priorité la plus élevée) et 9 (priorité la plus faible). La valeur implicite de XPRTY est déterminée par la classe du travail contenant l'activité. La priorité d'exécution d'une activité peut être modifiée par l'opérateur.

**Remarque :** Le système utilise les priorités d'exécution de 0 à 3. La valeur recommandée pour la priorité d'exécution des travaux utilisateur se situe donc entre 4 et 9 inclus.

**CPTIME**

Ce paramètre indique le temps maximum autorisé d'utilisation du processeur central par l'activité, en millièmes de minute. Le dépassement de la valeur indiquée provoque un arrêt anormal du travail (code de gravité 3) et l'émission du code retour TIMEOUT. La valeur implicite (99999999) signifie que le temps processeur n'est pas limité.

**ELAPTIME**

Ce paramètre spécifie le temps maximum entre le début et la fin d'exécution d'une activité. L'unité de temps est la minute. La valeur implicite (9999) signifie que le temps d'exécution de l'activité n'est pas limité. Le dépassement du temps indiqué provoque l'arrêt anormal du travail (code de gravité 3) et l'émission du code retour TIMEOUT.

**LINES**

Ce paramètre indique le nombre maximum de lignes pouvant être écrites dans chaque fichier SYSOUT par l'activité. Le dépassement de cette limite provoque l'arrêt anormal du travail et l'émission du code retour ERLMOV.

La valeur implicite (99999999) signifie que le nombre de lignes n'est pas limité.

**Remarque :** Si plusieurs exemplaires d'un fichier SYSOUT sont générés (voir le paramètre COPIES des ordres SYSOUT, WRITER et OUTVAL), le premier exemplaire compte seul pour le paramètre LINES.

**DUMP**

Ce paramètre indique si un listage de vidage doit être généré en cas d'arrêt anormal de l'activité et si oui, en spécifie le contenu.

NO	Valeur implicite. Pas de vidage.
DATA	Vidage des segments de données uniquement.
ALL	Vidage de tous les segments de l'activité.
PRIVATE	Vidage des segments du process uniquement (type 3).

**DEBUG**

Ce paramètre indique que l'activité doit être exécutée en mode mise au point. La valeur implicite est le mode normal. En environnement de traitement par lots, les commandes PCF d'aide à la mise au point proviennent du fichier d'entrée séquentiel spécifié ; en environnement IOF (interactif), elles proviennent de la console utilisateur.

**OPTIONS**

Ce paramètre spécifie une chaîne de 4096 caractères maximum, devant être lue par un programme utilisateur lors de l'exécution du module chargeable. Pour plus de détails sur l'utilisation du paramètre OPTIONS, voir le guide utilisateur relatif au langage concerné (FORTRAN, etc.).

**REPEAT**

Ce paramètre spécifie l'utilisation de la relance sur point de reprise : en cas de blocage du système ou d'arrêt anormal de l'activité, l'activité peut être relancée soit à partir du début, soit à partir du point de reprise défini.

Une activité définie avec l'option REPEAT, et devant utiliser un fichier spécifié par "nom-fichier-externe\*" ne peut pas être relancée sur point de reprise après un blocage du système ou un arrêt anormal de son exécution.

Lorsqu'une activité est définie sans l'option REPEAT, elle n'a implicitement pas de points de reprise et ne peut pas être relancée. Les appels à des points de reprise sont alors considérés comme des ordres fictifs ; il reste cependant possible de relancer le travail en entier (voir l'option REPEAT de l'ordre \$JOB).

**MAXMEM**

Lorsque ce paramètre autodéfini est spécifié, le système limite la place disponible en mémoire pour l'activité à la valeur de l'espace d'exécution déclaré (implicite ou déclaré dans SIZE), et ce, pour toute la durée d'exécution de l'activité. L'espace d'exécution déclaré est donc réservé à l'activité, mais l'activité ne peut à aucun moment occuper davantage de place en mémoire.

L'utilisation de ce paramètre est à réserver pour l'évaluation des performances de l'activité, lorsqu'il s'agit de trouver l'espace d'exécution déclaré optimum (voir le *Guide utilisateur JCL*).

**MINMEM**

Ce paramètre s'applique uniquement aux travaux de transmission. Lorsque ce paramètre autodéfini est spécifié, le système garantit à l'activité, pendant toute la durée de son exécution, une place correspondant à la valeur de l'espace d'exécution déclaré.

***Exemples :***

```
STEP MJPROG, TEMP;
```

Le module chargeable MJPROG est spécifié pour exécution. Il se trouve dans la bibliothèque temporaire standard TEMP.LMLIB.

```
STEP DMPROG, (NEW.DMLMLIB, DEVCLASS=MS/M500, MEDIA=B016);
```

Le module chargeable DMPROG est spécifié pour exécution. Il se trouve dans la bibliothèque NEW.DMLMLIB, qui n'est pas cataloguée et est implantée sur le volume B016 de l'unité de disques MSU0555.

```
STEP COBSTEP, NLIB.TEST, DUMP=DATA, REPEAT;
```

Le module chargeable COBSTEP, rangé dans la bibliothèque NLIB.TEST, est spécifié pour exécution. En cas d'arrêt prématuré de l'activité, il y aura vidage des segments de données. En cas de blocage du système, l'activité sera relancée depuis le début (ou depuis le dernier point de reprise, s'il y en a).

## 4.32 \$SWINPUT

### 4.32.1 Fonction

Spécifie une autre provenance pour les entrées (passage du train d'entrées ou du fichier en cours à un autre fichier).

### 4.32.2 Rubrique

Rubrique de description de travail, d'entrées ou de données.

### 4.32.3 Format

```
$SWINPUT
{[INFILE=](description-fichier-d'entrée-séquentiel)
nom-unité-bibliothèque [INLIB=](description-bibliothèque-d'entrée)}
{[ANSWERS=( 'chaîne105' ['chaîne105'...])}
{[ 'chaîne105' ] END='chaîne105'
};
```

### 4.32.4 Description

Le symbole "dollar" (\$) est obligatoire et doit être le premier caractère de l'article.

L'ordre \$SWINPUT est reconnu et traité par le lecteur des entrées. Il spécifie une autre provenance pour les entrées, qui viennent dès lors non plus du train d'entrées ou du fichier en cours, mais du fichier désigné. Du point de vue logique, ceci équivaut à la suppression des ordres \$SWINPUT du train d'entrées par le lecteur des entrées, et au remplacement de chacun par le contenu du fichier qu'il spécifie.

Les fichiers temporaires et les fichiers permanents catalogués dans des catalogues privés ne doivent pas être utilisés, sauf si le train d'entrées est soumis au moyen d'un ordre RUN. Un ordre \$SWINPUT peut figurer n'importe où dans une rubrique de description de travail, d'entrées ou de données. Il peut être imbriqué dans un ordre JCL, mais non dans un ordre du lecteur des entrées (\$JOB, \$ENDJOB, \$INPUT, \$ENDINPUT, \$DATA, \$ENDDATA, \$SWINPUT).

Le fichier spécifié par l'ordre \$SWINPUT ne doit pas contenir d'ordre \$ENDJOB ou \$ENDDATA.

Lorsqu'un ordre \$SWINPUT figure dans une rubrique de description d'entrées ou de données, le paramètre CKSTAT doit être spécifié dans l'ordre \$INPUT ou \$DATA concerné. La détection de la fin du fichier spécifié par \$SWINPUT provoque un retour au train d'entrées contenant l'ordre \$SWINPUT.

Au lieu de spécifier un fichier, un ordre \$SWINPUT peut citer une console. Lorsque le travail a été généré par un autre travail au moyen d'un ordre RUN, la console concernée est celle qui a soumis ce dernier. Dans tous les autres cas, la console est celle où le travail est soumis, c'est-à-dire celle de l'opérateur principal, de l'opérateur RBF ou de l'utilisateur IOF.

Un train d'entrées peut comporter n'importe quel nombre d'ordres \$SWINPUT, mais pas plus de 10 niveaux d'imbrication. Les fichiers spécifiés dans un ordre \$SWINPUT peuvent être affectés au moyen des paramètres INFILE2, INFILE3 et INFILE4 de l'ordre RUN.

L'ordre \$SWINPUT est utile dans un environnement interactif.

#### 4.32.5 Paramètres obligatoires

Les seuls paramètres obligatoires sont ceux qui spécifient le fichier duquel doivent provenir les entrées. Ce fichier peut être spécifié comme un fichier d'entrée séquentiel, une unité de bibliothèque d'entrée ou une console.

##### **description-fichier-d'entrée-séquentiel**

Ce groupe de paramètres peut être précédé par "INFILE=". Il désigne un fichier séquentiel, dont le contenu viendra remplacer l'ordre \$SWINPUT dans le train d'entrées. "description-fichier-d'entrée-séquentiel" est décrit au chapitre 3.

##### **description-bibliothèque-d'entrée**

Ce groupe de paramètres peut être précédé par "INLIB=". Il désigne une bibliothèque d'entrée dont le sous-fichier "nom-unité" sera sélectionné. Le contenu de cette unité viendra remplacer l'ordre \$SWINPUT dans le train d'entrées. "description-bibliothèque-d'entrée" est décrit au chapitre 3.

##### **CONSOLE**

Ce paramètre indique que les entrées doivent provenir d'une console (voir la description de l'ordre).

Elles sont alors effectuées de l'une des façons suivantes :

1. Option ANSWERS

La chaîne de caractères "chaîne105" précédant l'option ANSWERS spécifie le message de sollicitation envoyé à la console pour demander l'entrée voulue. La chaîne de caractères suivant ANSWERS définit les entrées possibles. L'opérateur de la console a droit à trois essais pour fournir l'une de ces réponses avant l'arrêt prématûre du travail. Une réponse correcte est insérée dans le train d'entrées à la place de l'ordre \$SWINPUT. La première réponse correcte constitue dans ce cas la seule nouvelle entrée.

Lorsque l'option ANSWERS et les réponses correctes sont omises, toute réponse ne dépassant pas la longueur d'un article est acceptée.

2. Option END

Plusieurs lignes d'entrée sont attendues de la console. La chaîne de gauche est le message de sollicitation initial, envoyé à la console pour demander la première ligne d'entrée. La chaîne suivante est le message sollicitant la seconde ligne et les lignes suivantes de l'entrée. Lorsque la seconde chaîne est omise, sa valeur implicite est celle de la première chaîne. Ces deux paramètres sont positionnels. La chaîne spécifiée avec "END=" est la ligne de terminaison indiquant la fin de l'entrée. Contrairement à l'option ANSWERS, les données introduites par l'opérateur ne sont pas limitées à une réponse choisie parmi un certain nombre de réponses correctes. Les réponses données avant la ligne de terminaison forment les nouvelles données d'entrée, qui sont insérées à la place de l'ordre \$SWINPUT dans le train d'entrées initial.

## 4.33 SYSOUT

### 4.33.1 Fonction

Demande l'utilisation du mécanisme SYSOUT pour un fichier à écrire dans le cadre de l'activité en cours, et sollicite l'éditeur des sorties pour l'impression du contenu de ce fichier, appelé fichier SYSOUT. L'ordre SYSOUT permet en outre de changer certaines options standard du mécanisme SYSOUT pour le traitement du fichier SYSOUT.

**Remarque :** Un fichier bibliothèque ou séquentiel permanent associé au nom de fichier interne spécifié s'appelle "fichier SYSOUT permanent". Si ce nom de fichier interne n'a pas été affecté par un ordre ASSIGN dans l'activité en cours, le système utilise un sous-fichier SYSOUT standard.

### 4.33.2 Rubrique

Description d'activité.

### 4.33.3 Format

```

SYSOUT nom-fichier-interne
  [CLASS=identificateur1]
  [PRIORITY=chiffre1]
  [
    {JOB      } ]
  [
    {STEP     } ]
  [
    {WHEN= {IMMED    } }
  [
    {DEFER    } ]
  [
    {chiffres5} ]
  [
    {HOLD     } ]
  [
    {NHOLD} ]
  [NAME=identificateur8]
  [
    {BANNER   } ]
  [
    {NBANNER} ]
  [
    {JOB      } ]
  [
    {OUTPUT   } ]
  [
    {BANLEVEL= {COPY     } }
  [
    {MEMBER   } ]
  [BANINF=(alphanum12 [,alphanum12]...)]

```

```
[COPIES=chiffres3]
[DEVCLASS=classe-d'appareils [MEDIA=nom-volume]]
[DEST=[nom-hôte].nom-station]
[ {SLEW } ]
[ {NSLEW} ]
[ {DELETE } ]
[ {NDELETE} ];
```

#### 4.33.4 Description

L'ordre SYSOUT sert à déclencher le mécanisme SYSOUT pour l'écriture d'articles générés par le programme utilisateur. Il indique également à l'éditeur des sorties comment traiter le fichier SYSOUT, en remplaçant certains paramètres de sortie implicites relatifs aux fichiers SYSOUT et définis à la configuration du système, tels que :

- l'identification de la sortie,
- certaines caractéristiques de sortie,
- le moment auquel l'éditeur des sorties est sollicité.

##### Fichiers SYSOUT standard

Si aucun ordre ASSIGN n'affecte le nom de fichier interne dans l'activité en cours, le système affecte automatiquement un sous-fichier temporaire du fichier SYSOUT standard. Ce sous-fichier est supprimé une fois son contenu imprimé. Il est possible d'utiliser le paramètre WHEN pour indiquer le moment où l'éditeur des sorties doit imprimer le contenu du fichier.

Si un programme COBOL contient le suffixe -SYSOUT dans sa clause SELECT...ASSIGN (voir la rubrique FILE-CONTROL dans le *Manuel de référence COBOL*), l'ordre SYSOUT peut être omis, sauf s'il y a des paramètres à remplacer. Le mécanisme SYSOUT est alors automatiquement appelé, le nom de fichier indiqué dans le programme est affecté au fichier SYSOUT standard, et les articles écrits dans le fichier sont imprimés par l'éditeur des sorties, implicitement sollicité en fin d'exécution du travail.

##### Fichiers SYSOUT permanents

L'ordre SYSOUT demande au mécanisme SYSOUT d'écrire les articles dans le fichier permanent associé au nom de fichier interne par un ordre ASSIGN dans l'activité en cours. L'éditeur des sorties est sollicité à un moment déterminé par la valeur du paramètre WHEN pour imprimer le contenu du fichier (voir également ci-après le cas où WHEN=DEFER).

Lorsqu'aucun paramètre SYSOUT n'est spécifié pour le fichier, mais qu'un programme COBOL contient le suffixe -SYSOUT dans sa clause SELECT...ASSIGN, le mécanisme SYSOUT est utilisé pour écrire des articles dans le fichier, mais le contenu du fichier n'est pas imprimé ; tel est également le cas s'il existe un ordre SYSOUT avec le paramètre WHEN=DEFER. Dans les deux cas, il faut spécifier ultérieurement un ordre WRITER - normalement dans un autre travail - pour imprimer le contenu du fichier.

Pour plus de précisions sur le mécanisme SYSOUT et l'éditeur des sorties, voir le *Guide utilisateur JCL*.

### Commentaires et restrictions

- L'ordre SYSOUT ne peut être utilisé pour le sous-fichier contenant le JOR. Dans ce cas, l'ordre approprié est OUTVAL (voir paragraphe 4.24).
- L'ordre DEFINE peut être utilisé pour remplacer des paramètres d'édition des sorties tels que les définitions de marges et de hauteur de feuillet. En ce qui concerne les fichiers SYSOUT permanents écrits au moyen du mécanisme SYSOUT (voir commentaire suivant), tous les paramètres d'édition implicites ou spécifiés par DEFINE sont incorporés à la structure du fichier ; les valeurs des paramètres d'édition sont donc automatiquement appliquées à l'impression du fichier lancée par l'ordre WRITER, sauf si le paramètre FPARAM est spécifié dans cet ordre (voir l'ordre WRITER au paragraphe 4.37).
- Lorsque le système réserve de la place à un fichier SYSOUT standard, une longueur d'article supérieure à 600 octets est établie. C'est une condition indispensable au fonctionnement du mécanisme SYSOUT. De même, lorsque l'utilisateur réserve de la place à un fichier SYSOUT permanent, l'ordre PREALLOC doit spécifier une longueur d'article supérieure à 600 octets avec RECFORM=VB pour que le mécanisme SYSOUT puisse être utilisé ; cette indication est indépendante de la longueur d'article déclarée dans le programme utilisateur (fichiers SYSOUT uniquement). Si la place nécessaire à un fichier SYSOUT permanent est réservée au moyen de l'ordre ALLOCATE, la longueur d'article voulue peut être déclarée dans l'ordre DEFINE. Si l'utilisateur veut écrire un fichier avec une longueur d'article inférieure à 600 octets, le mécanisme SYSOUT ne peut pas être utilisé pour éditer le fichier ; dans ce cas, l'ordre SYSOUT édite le fichier conformément aux paramètres de sortie et d'édition applicables (implicites et/ou indiqués par SYSOUT et/ou DEFINE) ; si l'ordre WRITER est spécifié ultérieurement pour ce fichier, ces paramètres doivent être respécifiés. Un ordre SYSOUT spécifié avec WHEN=DEFER est sans objet pour un fichier dont la longueur d'article est inférieure à 600 octets.

### 4.33.5 Paramètres

#### 4.33.5.1 Paramètre obligatoire

##### **nom-fichier-interne**

Identifie le fichier SYSOUT. Si aucun ordre ASSIGN correspondant ne figure dans l'activité en cours, il y a affectation automatique d'un sous-fichier du fichier SYSOUT standard.

#### 4.33.5.2 Paramètres facultatifs

##### **CLASS**

Ce paramètre permet à l'utilisateur de classer chaque fichier de sortie généré en fonction des besoins d'un groupe de sorties particulier. La valeur associée est une lettre. A chaque classe de sortie correspond une priorité de sortie implicite (voir le paramètre PRIORITY).

Si le paramètre CLASS n'est pas spécifié, sa valeur implicite est C. Pour les sorties locales, la valeur implicite de CLASS peut être modifiée soit à la génération du système, soit au moyen de la commande CATMAINT "MDS" (**MoDify Station**).

Le système des classes de sortie donne à l'opérateur un moyen de contrôle sur les sorties : si, par exemple, les sorties à imprimer sur un type de papier donné sont groupées dans la même classe de sortie, il peut faire en sorte que toutes les sorties de cette classe soient imprimées en même temps, réduisant ainsi les manipulations de papier.

##### **PRIORITY**

Ce paramètre a pour valeur un nombre entier compris entre 0 et 7, qui indique la priorité de sélection du fichier SYSOUT par l'éditeur des sorties (0 correspondant à l'urgence la plus élevée et 7 à la plus faible). PRIORITY spécifie donc l'ordre d'impression d'un fichier SYSOUT par rapport aux autres fichiers SYSOUT à imprimer par l'éditeur des sorties.

Les fichiers SYSOUT sont rangés en file d'attente des sorties de la priorité la plus élevée à la plus basse et, pour une priorité donnée, suivant le principe du "premier entré-premier sorti" (FIFO). La priorité implicite d'une sortie dépend de la classe de sortie spécifiée (ou implicite). Les correspondances sont les suivantes :

A	B	C	D	E	F à Z
1	2	3	4	5	6

Ces valeurs implicites peuvent être modifiées soit à la configuration du système, soit au moyen de la commande opérateur MOC. Pour les stations RBF, elles peuvent être modifiées au moyen de la commande CATMAINT "MDS". Noter que les nombres 0 et 7 ne sont pas utilisés dans ces valeurs implicites, sauf si l'utilisateur les a spécifiés au moyen de la commande GCL "MDC OC" ou des utilitaires CATMAINT ou CONFIG.

**Remarque :** La priorité de sortie indiquée par ce paramètre ne correspond ni à la priorité de planification indiquée par l'ordre \$JOB, ni à la priorité d'exécution d'activité indiquée par l'ordre STEP.

### WHEN

Ce paramètre indique le moment auquel l'éditeur des sorties est sollicité pour imprimer la sortie. Il peut prendre l'une des valeurs suivantes :

JOB	Sollicitation en fin d'exécution du travail. C'est la valeur implicite.
STEP	Sollicitation en fin d'exécution de l'activité. Utile pour les travaux à plusieurs activités, l'éditeur des sorties pouvant alors s'exécuter en même temps que d'autres activités du même travail.

**Remarque :** Si l'activité est relançable, WHEN=IMMED ou "chiffres5" est remplacé par WHEN=STEP, afin de ne pas imprimer deux fois la même chose. Si le travail est relançable, la mise en file d'attente de sortie est différée jusqu'à la fin du travail.

IMMED Sollicitation à la fermeture du fichier SYSOUT. Permet à l'éditeur des sorties de commencer à s'exécuter avant la fin de l'activité, si les conditions le permettent.

DEFER Uniquement applicable à un fichier SYSOUT permanent. Pas de sollicitation de l'éditeur des sorties. L'impression du fichier doit être demandée par un ordre WRITER, normalement dans un travail différent, à un moment ultérieur. Lorsque DEFER est spécifié, aucun des autres paramètres facultatifs n'est pris en compte, sauf DEVCLASS et MEDIA.

chiffres5 S'applique uniquement aux fichiers SYSOUT publics.

Un fichier SYSOUT distinct est généré après chaque sortie de ce nombre de pages. Une fois la constitution d'un tel fichier terminée, l'éditeur des sorties peut le traiter en fonction de ses priorité et classe de sortie.

Par exemple, si WHEN=100, chaque lot de 100 pages de sortie constitue un fichier SYSOUT distinct, que l'éditeur des sorties peut traiter aussitôt qu'il est complet.

Si l'activité est relançable, WHEN=IMMED ou "chiffres5" est remplacé par WHEN=STEP, afin de ne pas imprimer deux fois la même chose. Si le travail est relançable, la mise en file d'attente de sortie est différée jusqu'à la fin du travail.

**Remarque :** Le délai entre la sollicitation de l'éditeur des sorties et le commencement de l'impression dépend directement de la charge en cours de l'éditeur des sorties et de la priorité associée au fichier SYSOUT. Même si IMMED est spécifié, la sortie peut toujours être repoussée au-delà de l'exécution de l'activité ou du travail.

**HOLD**

Lorsque ce paramètre est spécifié, l'éditeur des sorties ne peut pas sélectionner le fichier SYSOUT pour impression tant que l'opérateur n'a pas émis d'ordre RO. HOLD a pour effet de suspendre temporairement la sortie du contenu d'un fichier SYSOUT rangé en file d'attente de sortie (liste, par ordre de priorité, des fichiers attendant leur traitement par l'éditeur des sorties). Noter que le paramètre HOLD prend effet après la sollicitation de l'éditeur des sorties ; il est donc indépendant du paramètre WHEN. Lorsque HOLD est spécifié, le paramètre NAME doit également l'être, afin que l'opérateur puisse identifier la sortie correspondant à l'ordre SYSOUT.

**NHOLD**

Ce paramètre indique que le contenu du fichier SYSOUT sortira sans intervention de l'opérateur. Il peut être utilisé pour le fichier en cours, afin de supplanter un paramètre HOLD spécifié dans un ordre OUTVAL précédent.

**NAME**

Ce paramètre spécifie un nom symbolique à associer à la sortie d'un fichier SYSOUT générée par l'éditeur des sorties. Ce nom peut être utile en cas d'arrêt anormal de l'impression, et peut figurer dans diverses commandes opérateur. Il peut comporter jusqu'à huit caractères (lettres, chiffres, trait d'union, caractère de soulignement) et doit commencer par une lettre. L'utilisation de NAME est fortement recommandée si HOLD est spécifié.

**Remarque :**

Le système fournit automatiquement un indice unique pour chaque unité de sortie, que NAME soit spécifié ou non. Le nom symbolique et l'indice peuvent être affichés sur la console opérateur, au moyen de l'ordre DO. En revanche, toutes les sorties de vidage (générées, par exemple, par la fonction DUMP COBOL) se voient attribuer un nom spécial par le système, et aucune valeur indiquée pour NAME n'est prise en compte.

**BANNER**

Ce paramètre rétablit l'effet d'une spécification du paramètre BANNER dans l'ordre OUTVAL.

**NBANNER**

Ce paramètre indique que les pages de bannière ne sont pas à imprimer.

**BANINF**

Ce paramètre permet à l'utilisateur de remplacer sur les pages de bannière les quatre rubriques standard utilisées par le système : numéro de passage (RON), nom d'utilisateur, nom de travail et compte.

Chaque valeur de remplacement peut comporter jusqu'à douze caractères. Si n valeurs sont mentionnées, n étant inférieur ou égal à 4, les n dernières rubriques sont remplacées. Il est impossible de spécifier des valeurs vides.

**Exemple :**

RON:X123, nom-utilisateur: MAXI, nom-travail: PRIM, compte: CAP

Si BANINF=(MINI,SEC,PES), les rubriques suivantes seront imprimées :

X123 MINI SEC PES

**BANLEVEL**

Permet de demander l'insertion de pages de bannière entre chaque unité ou exemplaire d'une sortie, ou de ne les faire apparaître que dans certains cas (changement de travail, de sortie, etc.).

Noter que les pages de bannière sont plus détaillées lorsqu'elles concernent uniquement une unité, un exemplaire ou une sortie. Le nom de la sortie, l'identification du fichier, le nom de l'unité ou le numéro d'exemplaire sont indiqués dans les cas appropriés.

Toute sortie placée en file d'attente soit immédiatement, soit en fin d'activité, soit avec l'option WHEN=IMMED, et pouvant donc être imprimée séparément du JOR et de JOBOUT, comporte des pages de bannière de niveau exemplaire au début et à la fin.

Pour les sorties placées en file d'attente en fin de travail comme le JOR et JOBOUT, les pages de bannière de niveau travail ne sont insérées qu'en cas de changement de travail ou de changement des valeurs spécifiées dans BANINF, même si plusieurs exemplaires ont été demandés.

Noter que BANLEVEL=MEMBER ne s'applique pas aux sorties envoyées sur un autre site ; dans ce cas, les différentes unités sont en effet regroupées par le mécanisme de désynchronisation du serveur.

BANLEVEL=JOB demande l'insertion de pages de bannière sur détection par le système d'un changement de travail. Les différentes sorties du travail ne sont toutefois pas regroupées ; il est donc recommandé de ne pas utiliser cette option pour les sorties placées en file d'attente avant la fin du travail.

**COPIES**

Ce paramètre indique le nombre d'exemplaires du fichier SYSOUT devant être générés par l'éditeur des sorties pour cet ordre. La valeur maximum est 99 et la valeur implicite 1.

**DEVCLASS**

Ce paramètre spécifie la classe et les caractéristiques de l'appareil de sortie (imprimante, par exemple). La classe d'appareils implicite dépend de l'installation.

**MEDIA**

Pour une imprimante, "nom-volume" précise le jeu de caractères d'impression et le type de papier sous la forme d'une valeur de six caractères :

- caractères 1 et 2 = jeu de caractères
- caractères 3 à 6 = numéro de papier

La valeur implicite dépend de l'installation.

## **DEST**

Ce paramètre est applicable sous RBF. Il indique la station RBF ou l'hôte à distance (ou encore une station dépendant de cet hôte à distance) auquel la sortie doit être envoyée. Si DEST n'est pas spécifié, la sortie est envoyée à la station émettrice du travail. La classe d'appareil par défaut, l'appareil, la classe d'appareil et le support sont ceux de la station à laquelle la sortie est associée.

## **SLEW**

S'il existe des instructions de saut de papier, elles sont exécutées à l'impression du fichier. Cette valeur est implicite à défaut d'ordre OUTVAL antérieur avec le paramètre NSLEW.

## **NSLEW**

Toute indication de saut de papier est transformée en interligne simple à l'impression du fichier par l'éditeur des sorties.

## **DELETE**

Ce paramètre s'applique uniquement à une unité de bibliothèque SYSOUT permanente. Lorsque DELETE est spécifié, le sous-fichier associé au nom de fichier interne est supprimé une fois les sorties imprimées. Il convient d'utiliser ce paramètre avec prudence, l'éditeur des sorties ne vérifiant pas si d'autres demandes de sortie (par SYSOUT ou WRITER) sont en attente pour ce fichier. Normalement, c'est-à-dire si DELETE ne figure ni dans l'ordre SYSOUT en cours, ni dans un ordre OUTVAL antérieur, il n'y a pas suppression d'unités de bibliothèque.

Si les sorties concernées sont annulées par une commande CO [STRONG], les sous-fichiers non entièrement imprimés par l'éditeur des sorties ne sont pas supprimés.

## **NDELETE**

Ce paramètre s'applique uniquement à une unité de bibliothèque SYSOUT permanente. Il est implicite si aucun ordre OUTVAL antérieur n'a spécifié DELETE. Si NDELETE est spécifié, le sous-fichier n'est pas supprimé après impression. Ce paramètre peut servir à supplanter un paramètre DELETE qui figurait dans un ordre OUTVAL antérieur pour le fichier SYSOUT en cours.

### ***Exemple 1 :***

```
SYSOUT LIST1, WHEN=STEP, NAME=HAJ;
```

Indique à l'éditeur des sorties qu'il doit imprimer le fichier SYSOUT portant le nom de fichier interne LIST1 à la fin de l'activité en cours. Le nom HAJ est associé à la sortie.

***Exemple 2 :***

```
SYSOUT HJPRINT, BANINF=(NJOB, PROJX);
```

Le contenu du fichier portant le nom de fichier interne HJPRINT sera imprimé. Les rubriques NJOB et PROJX figureront sur les pages de bannière à la place, respectivement, du nom de travail et du nom de projet. Le numéro de passage et le nom d'utilisateur à imprimer restent inchangés.

***Exemple 3 :***

```
ASSIGN DMPRINT, MY.LIB, SUBFILE=TEMPA;  
SYSOUT DMPRINT, COPIES=2, DELETE;
```

Il y a impression de deux exemplaires du sous-fichier de bibliothèque TEMPA et suppression du sous-fichier après impression.

## 4.34 VALUES

### 4.34.1 Fonction

L'ordre VALUES fixe la valeur implicite de certains paramètres des ordres JCL qui le suivent.

### 4.34.2 Rubrique

Description de travail, séquence appelée par INVOKE ou séquence EXECUTE.

### 4.34.3 Format

```
VALUES [valeur-paramètre1 [valeur-paramètre2]... ]  
[mot-clé1=valeur-mot-clé1 [mot-clé2=valeur-mot-clé2]...] ;
```

### 4.34.4 Description

L'ordre VALUES fixe des valeurs devant se substituer aux paramètres symboliques spécifiés dans une rubrique de description de travail. Ces valeurs peuvent aussi être des valeurs implicites de paramètres, ce qui permet de les omettre dans la liste du paramètre VALUES d'un ordre INVOKE, EXECUTE ou RUN, ou parmi les valeurs spécifiables dans une commande opérateur EJR. L'ordre VALUES doit se trouver dans la séquence JCL appelée par INVOKE, EXECUTE, RUN ou la commande opérateur EJR. Les valeurs implicites s'appliquent à tous les ordres JCL qui suivent jusqu'au prochain ordre VALUES/MODVL, ou bien jusqu'à la fin de la rubrique de description de travail ou de la séquence d'ordres JCL en cours.

Une valeur de paramètre remplace "&i", "i" représentant la position de la valeur de paramètre dans l'ordre VALUES (1<i<99). Une valeur de mot clé remplace "&mot-clé".

Les valeurs de paramètres sont de type positionnel, alors que les valeurs de mots clés figurent dans des paramètres à mot-clé (voir chapitre 2).

L'ordre VALUES ne peut servir à fixer de valeurs implicites ni pour les ordres \$SWINPUT, ni pour un ordre VALUES ultérieur.

Un ordre VALUES annule un ordre VALUES précédent éventuel, ainsi que l'effet d'ordres MODVL précédents.

Pour les règles concernant le remplacement des paramètres dans les ordres JCL, voir le *Guide utilisateur JCL*.

#### 4.34.5 Paramètres facultatifs

##### Valeur-paramètrei

"valeur-paramètrei" définit la valeur implicite de la chaîne "&i" (1<=i<=99) dans une séquence d'ordres JCL appelée par INVOKE ou dans une description de travail. Cette valeur doit être conforme aux règles de syntaxe du ou des paramètres pour lesquels elle est définie ; elle peut cependant être une chaîne protégée (n'importe quelle suite de caractères entre apostrophes, toute apostrophe apparaissant à l'intérieur de la chaîne devant être doublée) de 128 caractères maximum. La position d'une valeur de paramètre par rapport aux autres valeurs de paramètres dans un ordre VALUES définit la valeur de i pour l'identification de la chaîne "&i" correspondante.

Lorsque la virgule est utilisée comme séparateur, les paramètres positionnels omis sont signalés par autant de virgules, sans être précédées chaque fois d'une valeur. Si l'espace est utilisé comme séparateur, un paramètre positionnel omis est signalé par un symbole "#". Il n'est pas nécessaire de signaler les paramètres omis à droite de la dernière valeur spécifiée par des virgules ou des symboles "#" (voir l'exemple 2 ci-après).

##### Valeur-mot-cléi

"valeur-mot-cléi" définit la valeur implicite destinée à remplacer "&" suivi de "mot-cléi" dans une séquence d'ordres JCL appelée par INVOKE ou dans une description de travail. Cette valeur doit être conforme aux règles de syntaxe du ou des paramètres pour lesquels elle est définie ; elle peut cependant être une suite protégée de 128 caractères maximum. La longueur maximum d'un mot-clé est de huit caractères.

##### *Exemple 1 :*

```
VALUES MYFILE, ,MS/M500, STAT=TEMPRY;
```

Cet ordre définit les valeurs implicites MYFILE et MS/M500 destinées à remplacer respectivement chaque occurrence des chaînes &1 et &3, ainsi que la valeur implicite TEMPRY destinée à remplacer chaque occurrence de la chaîne &STAT. Chaque occurrence de la chaîne &2 sera implicitement remplacée par une chaîne vide (autrement dit, ce paramètre ne sera pas pris en compte).

L'ordre suivant :

```
ASSIGN IFN&2A, &1, DEVCLASS=&3, MEDIA=ABC, &STAT;
```

deviendra donc l'ordre ci-dessous, à condition que les valeurs implicites spécifiées dans l'ordre VALUES ne soient pas remplacées (par celles d'un ordre RUN, par exemple) :

```
ASSIGN IFNA, MYFILE, DEVCLASS=MS/M500, TEMPRY;
```

***Exemple 2 :***

Il y a 10 paramètres positionnels (&1 à &10) et les valeurs p2, p4 et p6 doivent être spécifiées respectivement pour le second, le quatrième et le sixième paramètre (donc : &2, &4, et &6). Les quatre ordres VALUES ci-après sont équivalents :

```
VALUES , p2,, p4,, p6,,,,,;  
VALUES , p2,, p4,, p6;  
VALUES # p2, # p4 # p6 # # # #;  
VALUES # p2 # p4 # p6;
```

## 4.35 WRITER

### 4.35.1 Fonction

Demande l'utilisation de l'éditeur des sorties pour un fichier SYSOUT permanent existant.

### 4.35.2 Rubrique

Description de travail.

### 4.35.3 Format

```
WRITER
{ (description-fichier-d'entrée séquentiel)
  {
    [ {PART=(ai [:bi] [ai+1 [:bi+1]]...) } ]
    [
      [ {SELECT=( [ron [:indice ]... } ]
      [ nom-sortie } ]
  }
  (description-bibliothèque-d'entrée)
  {
    SUBFILES=( {nom-unité [nom-unité]... } }
    {nom-avec-astérisque [nom-avec-astérisque]... } )
  }
  [CLASS=identificateur1]
  [PRIORITY=chiffre1]
  [WHEN ={JOB } ]
  [ {IMMED} ]
  [ {HOLD } ]
  [ {NHOLD} ]
  [NAME = identificateur8]
  [ {BANNER } ]
  [ {NBANNER} ]
  [
    [ {JOB } ]
    [ {OUTPUT } ]
  ]
  [ BANLEVEL {COPY } ]
  [ {MEMBER } ]
}
```

```

[BANINF = (alphanum12 [,alphanum12] . . . )]

[COPIES =chiffres2]

[DEVCLASS =classe-d'appareils [MEDIA =nom-volume]]
[DEST [= nom-hôte].nom-station3]

[DATAFORM = {SSF      [REPORT = {chaîne2}] }]
[          {SARF      {ALL      } } ]
[          {DOF       } ]
[          {ASA       } ]

[NUMBER]

[ {FPARAM } ]
[ {NFPARAM} ]

[ {SLEW } ]
[ {NSLEW} ]

[ {DELETE } ]
[ {NDELETE} ]

[PRINTER = (options-imprimante)];
```

Le format d'"options-imprimante" est détaillé dans la description de l'ordre DEFINE.

#### 4.35.4 Description

L'ordre WRITER demande l'utilisation de l'éditeur des sorties pour un fichier SYSOUT permanent existant, enregistré sur disque ou bande. Ce fichier peut être soit un fichier UFAS séquentiel, soit un fichier bibliothèque, catalogué ou non ; il peut avoir été constitué ou non au moyen du mécanisme SYSOUT. L'ordre WRITER est souvent spécifié dans un travail autre que celui dans lequel le fichier a été créé.

Le mécanisme SYSOUT est expliqué de façon générale dans l'ordre SYSOUT (paragraphe 4-35), et plus précisément dans le guide utilisateur JCL (12UJ).

L'ordre WRITER permet à l'utilisateur de spécifier des valeurs non standard pour les paramètres d'édition des sorties (par exemple, définition des marges et densité d'impression) et les paramètres de traitement des sorties (par exemple, destination des sorties et nombre d'exemplaires).

Il est possible de sélectionner pour impression certaines pages du fichier SYSOUT ou certains sous-fichiers d'un fichier bibliothèque SYSOUT.

### Commentaires et restrictions

- Toute valeur de paramètre de traitement des sorties (DEST ou COPIES, par exemple) spécifiée au moyen, par exemple, d'un ordre SYSOUT, à l'écriture du fichier, n'est applicable qu'à l'activité pendant laquelle une écriture dans le fichier a eu lieu. Si l'utilisateur veut changer la valeur standard de l'un de ces paramètres, il doit spécifier la nouvelle valeur dans l'ordre WRITER en cours.
- Si le fichier à imprimer a été écrit en utilisant le mécanisme SYSOUT, aucun paramètre d'édition (PRINTER) spécifié dans l'ordre WRITER ne sera pris en compte. Autrement dit, les indications données à l'écriture du fichier restent implicitement applicables. Si l'utilisateur veut néanmoins remplacer ces indications initiales (soit implicites, soit correspondant à la valeur de MEDIA spécifiée dans l'ordre SYSOUT, soit fournies dans un ordre DEFINE), le paramètre FPARAM est obligatoire dans l'ordre WRITER.
- L'ordre WRITER ne peut figurer que dans une rubrique de description de travail, à l'extérieur de toute rubrique de description d'activité. L'ordre SYSOUT ne peut au contraire figurer qu'à l'intérieur d'une rubrique de description d'activité.
- L'exécution de l'éditeur des sorties pour imprimer le contenu du fichier indiqué n'est pas forcément synchrone avec celle du travail contenant l'ordre WRITER. Un fichier SYSOUT permanent ne doit donc pas être supprimé (par DEALLOC, par exemple) dans le même travail que celui où figure la demande d'impression le concernant (ni même dans les travaux ultérieurs, s'il ne s'est pas écoulé suffisamment de temps), car la suppression risque d'avoir lieu avant que l'impression soit terminée. C'est pour cette raison qu'un fichier temporaire ou une rubrique de description d'entrées n'est pas autorisé. Voir la description du paramètre DELETE dans l'ordre SYSOUT.

## 4.35.5 Paramètres

### 4.35.5.1 Paramètres obligatoires

Il faut spécifier soit une description de fichier d'entrée séquentiel, soit une description de bibliothèque d'entrée, mais pas les deux.

#### **description-fichier-d'entrée-séquentiel**

Identifie un fichier séquentiel (ou une unité de bibliothèque) dont le contenu doit être imprimé par l'éditeur des sorties. Le développement de ce groupe de paramètres est fourni au chapitre 3. Noter que cette description de fichier ne peut identifier qu'une seule unité de bibliothèque (paramètre SUBFILES).

Il est possible :

- d'imprimer seulement certaines pages (paramètre PART) ;
- d'imprimer plusieurs exemplaires (paramètre COPIES) ;

- de sélectionner le ou les états à imprimer (paramètre SELECT) si le fichier concerné a été généré par l'éditeur des sorties.

**Exemple :**

```
WRITER (.MYLIB, SUBFILE=S1), COPIES=6;
```

Le contenu de l'unité S1 de la bibliothèque cataloguée .MYLIB est imprimé six fois.

**description-bibliothèque-d'entrée**

Identifie la bibliothèque qui contient les unités (spécifiées dans le paramètre SUBFILES décrit ci-après) dont le contenu doit être imprimé par l'éditeur des sorties. Le développement de ce paramètre figure au chapitre 3.

**SUBFILES**

Ce paramètre s'applique uniquement aux fichiers bibliothèques. Il sélectionne pour impression les sous-fichiers voulus du fichier bibliothèque spécifié. Il est possible de sélectionner n'importe quel nombre de familles de sous-fichiers en utilisant un nom avec astérisque (voir le *Manuel de référence relatif à la maintenance des bibliothèques*).

Lorsque SUBFILES est spécifié, les paramètres PART, SELECT et COPIES doivent être omis.

**Exemple :**

```
WRITER (.MYLIB), SUBFILES=(S1, S2, S3);
```

Demande d'impression des unités S1, S2 et S3 de la bibliothèque cataloguée .MYLIB.

Pour demander l'impression d'une seule unité de bibliothèque, utiliser plutôt "description-fichier-d'entrée-séquentiel".

#### 4.35.5.2 Paramètres facultatifs

Pour une description détaillée des paramètres non mentionnés ci-dessous (paramètres de traitement des sorties CLASS, PRIORITY, HOLD, NHOLD, NAME, BANNER, NBANNER, BANINF, DEVCLASS, MEDIA, DEST, SLEW, NSLEW, DELETE, NDELETE), voir l'ordre SYSOUT.

**PART**

Ce paramètre permet à l'utilisateur de définir la ou les parties du fichier (ou du sous-fichier de bibliothèque) spécifié à imprimer. Chaque paire de valeurs (ai:bi) indique la partie voulue par la première et la dernière page à imprimer. Toute valeur fournie doit être un chiffre ou une suite de chiffres. La valeur implicite de "bi" est "ai".

Une liste de paires de valeurs séparées par une virgule, doit apparaître en ordre croissant des numéros de pages ; la valeur de "ai" doit donc être inférieure ou égale à celle de "bi" et celle de "bi" doit être inférieure à celle de "aj" (j correspondant à i+1). Dans la dernière paire de valeurs, "bi" peut prendre la valeur "\$" pour indiquer que la limite supérieure est la fin du fichier. Dans ce cas, "ai" est nécessaire et ne peut pas prendre la valeur "\$".

**Exemple :**

PART=(4:7, 9, 11:11, 13:\$)

Cet exemple indique les pages 4, 5, 6, 7, 9, 11 et de 13 à la fin du fichier. Noter que la deuxième page indiquée aurait pu être désignée par 9:9 et la troisième par 11. Noter également que la paire "13:\$" est correcte même si 13 est la dernière page du fichier. La dernière paire de valeurs ne peut cependant pas être désignée par "\$:\$" ou "\$".

**SELECT**

Ce paramètre permet de sélectionner des états relatifs à un travail en spécifiant soit un numéro de passage (RON), soit à l'intérieur du RON, un indice ou un nom de sortie. Il s'applique seulement quand le fichier spécifié a été généré par l'éditeur des sorties.

Lorsque SELECT=ron, toutes les sorties du travail portant ce numéro de passage sont imprimées.

Lorsque SELECT=ron:nom-de sortie, c'est la sortie spécifiée du travail ayant le numéro de passage indiqué qui est imprimée.

Lorsque SELECT=ron:indice, c'est la sortie avec l'indice spécifié du travail ayant le numéro de passage indiqué qui est imprimée (voir le paramètre NAME de l'ordre SYSOUT pour une explication de l'indice de sortie).

**COPIES**

Ce paramètre indique le nombre d'exemplaires à imprimer. La valeur maximum est 99 et la valeur implicite 1. COPIES n'est pas pris en compte si SUBFILES ou PART est spécifié, ou si le fichier à traiter par WRITER a été créé par l'éditeur des sorties.

**WHEN**

Ce paramètre indique le moment auquel l'éditeur des sorties est sollicité pour l'impression. Les valeurs possibles sont les suivantes :

JOB	Sollicitation en fin d'exécution du travail. Valeur implicite.
IMMED	Sollicitation à l'exécution de l'ordre WRITER.

Noter que le délai entre la sollicitation de l'éditeur des sorties et le début de l'impression sur l'appareil de sortie dépend directement de la charge de l'éditeur des sorties et de la priorité de la classe de travaux associée au fichier SYSOUT. Même si IMMED est spécifié, l'impression peut être repoussée au delà de l'exécution du travail.

**DATAFORM**

Ce paramètre n'est applicable que si le mécanisme SYSOUT n'a pas été utilisé pour l'écriture du fichier. Il n'est pas pris en compte pour les fichiers SYSOUT en format SYSOUT. Le format spécifié dans DATAFORM prévaut sur le format calculé par WRITER. Il n'est utile de le spécifier que pour les sorties en format ASA, les articles en format SSF et DOF étant automatiquement reconnus par WRITER. Lorsque le fichier n'est ni en format SSF, ni en format DOF, SARF est le format implicitement retenu.

SSF	Les articles du fichier SYSOUT contiennent des données et un en-tête SSF. Valeur implicite.
SARF	Les articles du fichier SYSOUT contiennent uniquement des données.
DOF	Obligatoire pour l'impression d'un fichier SYSOUT permanent Mode 100. Il s'agit d'un format de données dépendant du type d'appareil.
ASA	Les articles du fichier SYSOUT contiennent des données et un en-tête ASA d'un octet (octet de saut). Ce premier caractère indique le type de saut à exécuter au cours du traitement de sortie de l'article. Pour plus de détails, voir le <i>Guide utilisateur concernant les périphériques autres que disques et bandes</i> .

**REPORT**

Ce paramètre s'applique aux fichiers créés par le programme d'édition COBOL.

chaîne_2	Lorsqu'une valeur de 2 caractères est spécifiée, elle devient le code d'identification d'article. Seuls les articles précédés de ce code sont pris en compte par WRITER. Le code d'identification d'article n'apparaît pas à l'impression. Les caractères non alphanumériques utilisés doivent figurer entre apostrophes. Si le paramètre REPORT est omis, tout code d'identification d'article existant apparaît (autrement dit, les articles sortent en entier).
ALL	Tous les articles du fichier sont imprimés, qu'ils aient ou non un code d'identification. Ce code n'apparaît pas à l'impression. ALL est la valeur implicite.

**ATTENTION**

Pour pouvoir utiliser la fonction de sélection d'article du programme d'édition COBOL, le fichier à imprimer doit être en format SSF. Il ne faut donc pas utiliser le mécanisme SYSOUT (à l'écriture du fichier) pour mettre le fichier en format SYSOUT (pour tous détails sur le mécanisme SYSOUT, voir l'ordre SYSOUT et le *Guide utilisateur JCL*).

**NUMBER**

Ce paramètre s'applique uniquement aux unités de bibliothèque origine en format SSF et non temporaires. Il indique que le numéro de ligne (de l'en-tête SSF) doit également être imprimé.

**FPARAM**

Ce paramètre n'est applicable qu'aux fichiers en format SYSOUT (édités avec le mécanisme SYSOUT). Il est obligatoire si les "paramètres-imprimante" (PRINTER) indiqués dans cet ordre WRITER doivent être appliqués. Si FPARAM est omis, les indications données à l'écriture du fichier sont appliquées même si PRINTER est spécifié dans cet ordre WRITER. Autrement dit, FPARAM sert à imposer d'autres paramètres d'édition.

**NFPARAM**

Ce paramètre spécifie le contraire de FPARAM : les paramètres d'édition spécifiés ne sont pas valables pour les fichiers déjà en format SYSOUT. Valeur implicite si FPARAM est omis.

**PRINTER**

Ces options sont applicables aux fichiers en format SYSOUT à condition que le paramètre FPARAM soit spécifié, et aux fichiers en format autre que SYSOUT. Les valeurs de paramètres fournies l'emportent sur les valeurs implicites (et, dans le cas de fichiers en format SYSOUT lorsque FPARAM est spécifié, sur les valeurs appliquées par DEFINE à la création du fichier).

Les options PRINTER permettent à l'utilisateur de spécifier des paramètres d'édition pour les fichiers à imprimer sur une imprimante ligne. Se reporter à l'ordre DEFINE pour la description des paramètres (mais noter que SLEW et NSLEW ne font pas partie du jeu de paramètres PRINTER - pour ces deux paramètres, voir l'ordre WRITER).

***Exemples :***

```
WRITER HJ.REPS, COPIES=3;
```

Cet ordre provoque l'impression du fichier SYSOUT permanent HJ.REPS en trois exemplaires.

```
WRITER NOV.FIGS, PRINTER=(PRDEN=8, FORMHT=80);
```

A condition que le fichier NOV.FIGS n'ait pas été écrit par le mécanisme SYSOUT, cet ordre WRITER en imprime le contenu, en fixant la densité d'impression à 8 lignes par pouce et la valeur de FORMHT à 80.

## 5. Ordres JCL étendus

Les ordres JCL étendus se présentent sous le même format que les ordres JCL de base, et obéissent aux mêmes règles de syntaxe. D'autre part, leur développement constituant généralement une activité, ils ne doivent jamais figurer dans une rubrique de description d'activité, à l'exception de SORTWORK et de GSORTWK.

Dans le tableau 5-1 ci-après, les ordres JCL étendus sont classés par ordre alphabétique. Veuillez noter que les ordres EXDIR, EXECUTE et RUN sont décrits au chapitre 4, avec les ordres JCL de base. La colonne "Ordre" contient le nom de l'ordre ; la colonne "Référence" donne le code du manuel où l'ordre concerné est présenté en détail (voir le tableau 5-2 pour la signification des codes), et la colonne "Fonction" en décrit brièvement l'utilisation.

**Tableau 5-1. Récapitulatif des ordres JCL Etendus (1/4)**

Ordre	Référence	Fonction
BINDER	BIUG	Active l'exécution des commandes Binder.
C	CLUG	Compile un programme origine C.
CATALOG	CATM	Inscrit un objet (fichier ou répertoire) dans un catalogue.
CATBUILD	CATM	Crée un catalogue privé.
CATCHECK	CATM	Vérifie la validité des structures de catalogues.
CATDELETE	CATM	Supprime un catalogue privé.
CATEXTD	CATM	Etend la dimension d'un catalogue.
CATLIST	CATM	Liste les structures de catalogues.
CATMAINT	SADM	Crée, modifie et supprime des informations dans le catalogue de site.
CATMODIF	CATM	Modifie la description d'un fichier dans un catalogue.
CATMOVE	CATM	Copie les structures d'un catalogue à un autre.
CBL	COUG	Compile un programme origine COBOL.
CMDMGT	GCPM	Compile les procédures origine GCL.
COMPARE	DMUT	Compare logiquement le contenu de deux fichiers et imprime les différences.
CREATE	DMUT	(1) Charge ou réorganise un fichier UFAS. (2) Copie un fichier IDS/II.
DEALLOC	DMUT	Libère l'espace réservé à un fichier bande catalogué, à un fichier disque ou à un fichier disquette.
DMLPROC	IDUG	Effectue le pré-traitement d'un programme COBOL contenant des ordres DML.
DUMPJRNL	TDSU	Extrait du journal Après les données privées d'un utilisateur TDS et les range dans le journal utilisateur.
FILALLOC	DMUT	Procède à une réservation statique de place pour n'importe quel type de fichier disque.
FILCHECK	DMUT	Vérifie qu'un fichier disque séquentiel ou bibliothèque peut être traité par les méthodes d'accès standard.
FILDUPLI	DMUT	Copie le contenu d'un fichier origine dans un autre fichier de même type.
FILLIST	DMUT	Liste les informations contenues dans certaines parties du label d'un fichier, ainsi que des informations de catalogage et d'utilisation concernant ce fichier.
FILMAINT	DMUT	(1) Modifie les données ou les structures de gestion à l'intérieur des articles physiques d'un fichier disque. (2) Procède au vidage des articles physiques d'un fichier disque ou bande natif.
FILMODIF	DMUT	(1) Change le nom et/ou la date de péréemption d'un fichier disque ou disquette. (2) Change le statut (catalogué ou non) d'un fichier disque ou bande. (3) Supprime des données dans un fichier disque ou disquette.

**Tableau 5-1. Récapitulatif des ordres JCL Etendus (2/4)**

Ordre	Référence	Fonction
FILREST	DMUT	Restaure le contenu d'un fichier disque à partir d'un fichier bande ou d'un fichier disque UFAS séquentiel.
FILSAVE	DMUT	Sauvegarde sur bande l'intégralité d'un fichier disque.
FILTFR	UFTG	Transfère vers un site distant un fichier qu'il peut créer, renommer ou supprimer.
FORMGEN	FOUG	Appelle l'utilitaire de génération de grilles.
FOR77	F7UG	Compile un programme origine FORTRAN77.
GPL	GPUG	Compile un programme origine natif en GPL.
GSORT	SMUG	Concatène jusqu'à 9 fichiers d'entrée et trie le fichier résultant sur des clés spécifiées dans un fichier séparé.
GSORTWRK	SMUG	Spécifie le type d'appareil (bande ou disque) et la quantité d'espace de travail à réservé lorsque l'utilitaire SORT est appelé de façon dynamique (par exemple, au moyen du verbe SORT) à partir d'un programme utilisateur.
JAGEN	TDSU	Appelle le processeur de maintenance du journal Après.
JASMAINT	FRFU	Active l'exécution des fonctions de maintenance du JAS.
LIBALLOC	DMUT	Réserve de la place pour les bibliothèques utilisateur.
LIBDELET	DMUT	Supprime une bibliothèque et libère l'espace qu'elle occupait, ou supprime le contenu de son répertoire et de ses unités, mais sans libérer l'espace qu'elle occupe.
LIBMAINT	LMRM	Appelle le processeur de maintenance de bibliothèque.
LINKER	LKUG	Procède à l'édition de liens d'une ou plusieurs unités compilées, qui forment ainsi un module chargeable exécutable.
MERGE	SMUG	Fusionne jusqu'à huit fichiers d'entrée suivant des clés spécifiées dans un fichier séparé.
MIRFIL	DMUT	Copie le contenu d'un disque MIRROR ALONE vers un disque FBO standard vide, et crée une paire de disques miroir.
MIRSTART	DMUT	Crée une paire de disques miroir à partir de deux disques FBO non miroir standard sans aucun fichier.
PASCAL	PAUG	Compile un programme origine natif en PASCAL.
PREALLOC	DMUT	Réserve de la place pour un fichier disque ou disquette, ou pour un fichier bande catalogué.
PRINT	DMUT	Imprime les articles d'un fichier.

**Tableau 5-1. Récapitulatif des ordres JCL Etendus (3/4)**

Ordre	Référence	Fonction
ROLLFWRD	TDSU	Restaure les fichiers jusqu'à un point de relance en utilisant le journal Après.
SETLIST	DMUT	Génère un état contenant le nom de tous les fichiers membres d'ensembles de fichiers et montrant la correspondance entre les fichiers d'entrée et les fichiers de sortie.
SHIFT	CATM	Met à jour le groupe de générations d'un fichier catalogué.
SORT	SMUG	Concatène jusqu'à neuf fichiers d'entrée et trie le fichier résultant sur des clés spécifiées dans un fichier séparé.
SORTIDX	DMUT	Trie et charge les index secondaires d'un fichier séquentiel indexé UFAS.
SORTWORK	SMUG	Spécifie le type d'appareil (bande ou disque) et la quantité d'espace de travail à réserver lorsque l'utilitaire SORT est appelé de façon dynamique (par exemple, au moyen du verbe SORT) dans un programme COBOL.
SYSMAINT	TDAG	Active l'exécution des commandes de maintenance du système.
TAPEPREP	DMUT	Dote les volumes bande d'un label et les formatent en vue de leur utilisation par un système d'exploitation GCOS, ou supprime les labels standard existants.
UNCAT	CATM	Supprime un objet dans un catalogue.
URINIT	GSOG	Met à jour le fichier système contenant les paramètres de gestion des périphériques autres que disques et bandes.
VOLCHECK	DMUT	Contrôle l'intégrité d'un volume disque.
VOLCOMP	DMUT	Compare deux volumes disque ou bande du même type et signale les différences.
VOLDUPLI	DMUT	Copie la totalité du contenu d'un volume disque ou bande sur un volume du même type.
VOLLIST	DMUT	Liste le contenu d'un volume disque ou bande natif.
VOLMAINT	DMUT	Affiche les articles physiques enregistrés dans le volume disque, disquette ou bande spécifié. Peut également modifier les articles physiques enregistrés dans un volume disque.
VOLMODIF	DMUT	(1) Signale les pistes défectueuses d'un disque sans supprimer les fichiers enregistrés sur ce disque. (2) Formate les zones libres d'un disque. (3) Remet à zéro le drapeau de réservation séquentielle sur les volumes disquette.

*Tableau 5-1. Récapitulatif des ordres JCL Etendus (4/4)*

Ordre	Référence	Fonction
VOLPREP	DMUT	Dote les volumes disque ou bande d'un label et les formate en vue de leur utilisation par un système d'exploitation GCOS, ou supprime les labels standard existants.
VOLPRINT	DMUT	Imprime tout ou partie du contenu d'un volume bande.
VOLREST	DMUT	Restaure le contenu d'un volume disque à partir d'un fichier bande.
VOLSAVE	DMUT	Sauvegarde le contenu d'un volume disque dans un fichier bande natif avec labels.
VSETLIST	ASMG	Génère la liste de tous les volumes appartenant à la configuration, indiquant l'espace occupé et l'espace disponible.

**Tableau 5-2. Explication des références bibliographiques du tableau 5-1**

Référence	Nom du manuel
ASMG	<i>Administration du gestionnaire d'espace disque - Guide de l'administrateur (36UF)</i>
BIUG	<i>BINDER - Guide de l'utilisateur (11UP)</i>
CATM	<i>Gestion des catalogues - Guide de l'utilisateur (35UF)</i>
CLUG	<i>Langage C - Guide de l'utilisateur (60UL)</i>
COUG	<i>COBOL 85 - Guide de l'utilisateur (06UL)</i>
DMUT	<i>Utilitaires de gestion de données (34UF)</i>
F7UG	<i>FORTRAN 77 - Guide de l'utilisateur (16UL)</i>
FSEG	<i>FSE (Editeur plein écran) - Guide de l'utilisateur (06UP)</i>
FOUG	<i>FORMS - Guide de l'utilisateur (15UJ)</i>
FRFU	<i>Reconstitution des fichiers - Guide de l'utilisateur (37UF)</i>
GCPM	<i>GCL - Guide du programmeur (36UJ)</i>
GPUG	<i>GPL - Guide de l'utilisateur (36UL)</i>
GSOG	<i>GCOS 7-V8 - Manuel d'exploitation (53US)</i>
IDUG	<i>IDS/II - Guide de l'utilisateur (12UD)</i>
IDUG	<i>IDS/II Etendu - Guide de l'utilisateur (07UD)</i>
LKUG	<i>LINKER (Editeur de liens) - Guide de l'utilisateur (10UP)</i>
LMRM	<i>Maintenance des bibliothèques - Manuel de référence (01UP)</i>
PAUG	<i>PASCAL - Guide de l'utilisateur (52UL)</i>
SADM	<i>GCOS 7-V8 - Manuel d'exploitation (54US)</i>
SMUG	<i>Tri et fusion - Guide de l'utilisateur (08UF)</i>
TDAG	<i>TDS - Guide de l'administrateur (32UT)</i>
TDSU	<i>TDS/COBOL - Manuel du programmeur (33UT)</i>
UFTG	<i>UFT - Guide de l'utilisateur (13UC)</i>

## A. Liste des ordres JCL et de leurs abréviations

Cette annexe contient la liste de tous les ordres JCL de base et étendus pour lesquels existent des abréviations standard. Les ordres sont classés par ordre alphabétique, chacun étant suivi de l'abréviation ou des abréviations autorisées.

Un exemple de description de travail avec les noms d'ordres sous forme intégrale, puis sous forme abrégée, figure à la fin de l'annexe.

- Remarques :**
1. Les noms d'ordres sont donnés sous leur forme intégrale dans le présent manuel, ainsi que dans toute la documentation GCOS 7.
  2. Il n'existe pas de forme abrégée pour tous les ordres JCL.

**Tableau A-1. Noms complets et abréviations des ordres JCL (1/3)**

<b>Nom complet</b>	<b>Abréviation(s)</b>
ALLOCATE	ALC
ASSIGN	ASG
ATTACH	ATT
CATALOG	CAT
CATBUILD	CBD
CATCHECK	CCK
CATDELETE	CDL
CATEXTD	CEXT
CATLIST	CLS
CATMAINT	CMN
CATMOVE	CMV
COBOL	CBL
COMMENT	COMM
COMPARE	CMP
CONSOLE	CNSL
CREATE	CR
DEALLOC	DALC
DEFINE	DEF
DMLPROC	DMLP
ENDDATA	EDATA
ENDINPUT	EIN
ENDJOB	EJOB
ENDSTEP	EST, ESTP

**Tableau A-1. Noms complets et abréviations des ordres JCL (2/3)**

<b>Nom complet</b>	<b>Abréviation(s)</b>
EXECUTE	EXC
FILCHECK	FCK
FILDUPLI	FDUP
FILMODIF	FMOD
FILREST	FRST
FILSAVE	FSV
FORTRAN77	F7C
INPUT	IN
INVOKE	IVK, IV
JASMAINT	MNJAS
LABEL	LBL
LIBALLOC	LALC
LIBDELET	LDL
LIBMAINT	LMN
LINKER	LINK, LK
MERGE	MRG
MESSAGE	MSG
MODIFY	MOD
MODVL	MVL
OUTVAL	OVL
PREALLOC	PALC
PREFIX	PFX

**Tableau A-1. Noms complets et abréviations des ordres JCL (3/3)**

<b>Nom complet</b>	<b>Abréviation(s)</b>
PRINT	PR
QASSIGN	QASG
RELEASE	RLS
REPLACE	REPL
REPORT	RPT
SFLIST	SFLS
SHIFT	SHF
SIZE	SZ
SORTFMT	SFMT
SORTIDX	SIDX
SORTWORK	SWK
STEP	STP
SWINPUT	SWI
SYSOUT	SYO
UNCAT	UNC
VALUES	VALUE, VL
VOLCHECK	VCK
VOLCOMP	VCMP
VOLDUPLI	VDUP
VOLPREP	VPRP
VOLREST	VRST
VOLSAVE	VSV
WRITER	WR

***Exemple de description de travail***

1. Noms d'ordres sous forme intégrale :

```
$JOB DUPONT, USER=PHILIPPE, PROJECT=UN;
  OUTVAL;
  OUTVAL COPIES=2;
    STEP TSTA, MST.CDM;
      ASSIGN IN99, .DAT99;
      ASSIGN SCR99, .ERR99;
    ENDSTEP;
  JUMP CONTINUE;
  JUMP END, SEV, LT, 3;
    WRITER .ERR99;
  END;
$ENDJOB;
```

2. Noms d'ordres sous forme abrégée :

```
$JOB DUPONT, USER=PHILIPPE, PROJECT=UN;
  OVL;
  OVL COPIES=2;
    STP TSTA, MST.CDM;
      ASG IN99, .DAT99;
      ASG SCR99, .ERR99;
    EST;
  JUMP CONTINUE;
  JUMP END, SEV, LT, 3;
    WR .ERR99;
  END;
$ENDJOB;
```



## **B. Conversion des caractères**

Cette annexe donne la liste de tous les codes EBCDIC (Extended Binary-Coded Decimal Interchange Code) avec la correspondance en code ASCII.

Elle fournit également les ordres de classement COBOL, la représentation graphique ou la signification correspondante sur DPS 7000.

## B.1 ASCII

ASCII										Indique l'absence de code de commande ou de symbole graphique									
EBCDIC					OC COBOL					EBCDIC					OC COBOL				
Code : Symbole										Code : Symbole									
00	00	1	NUL	00	00	1	ASCII	00	00	1	ASCII	00	00	1	ASCII	00	00	1	ASCII
01	01	2	SOH	01	01	2	EBCDIC	01	01	2	EBCDIC	01	01	2	EBCDIC	01	01	2	EBCDIC
02	02	3	STX	02	02	3	OC COBOL	02	02	3	OC COBOL	02	02	3	OC COBOL	02	02	3	OC COBOL
03	03	4	ETX	03	03	4	Code : Symbole	03	03	4	Code : Symbole	03	03	4	Code : Symbole	03	03	4	Code : Symbole
04	37	56	EOT	04	37	56	ASCII	04	37	56	ASCII	04	37	56	ASCII	04	37	56	ASCII
05	2D	46	ENQ	05	2D	46	EBCDIC	05	2D	46	EBCDIC	05	2D	46	EBCDIC	05	2D	46	EBCDIC
06	2E	47	ACK	06	2E	47	OC COBOL	06	2E	47	OC COBOL	06	2E	47	OC COBOL	06	2E	47	OC COBOL
07	2F	48	BEL	07	2F	48	Code : Symbole	07	2F	48	Code : Symbole	07	2F	48	Code : Symbole	07	2F	48	Code : Symbole
08	16	23	BSV	08	16	23	ASCII	08	16	23	ASCII	08	16	23	ASCII	08	16	23	ASCII
09	05	6	HTV	09	05	6	EBCDIC	09	05	6	EBCDIC	09	05	6	EBCDIC	09	05	6	EBCDIC
0A	25	38	LFV	0A	25	38	OC COBOL	0A	25	38	OC COBOL	0A	25	38	OC COBOL	0A	25	38	OC COBOL
0B	OB	12	VTV	0B	OB	12	Code : Symbole	0B	OB	12	Code : Symbole	0B	OB	12	Code : Symbole	0B	OB	12	Code : Symbole
0C	OC	13	FFV	0C	OC	13	ASCII	0C	OC	13	ASCII	0C	OC	13	ASCII	0C	OC	13	ASCII
0D	OD	14	CRV	0D	OD	14	EBCDIC	0D	OD	14	EBCDIC	0D	OD	14	EBCDIC	0D	OD	14	EBCDIC
0E	OE	15	SOV	0E	OE	15	OC COBOL	0E	OE	15	OC COBOL	0E	OE	15	OC COBOL	0E	OE	15	OC COBOL
0F	OF	16	SIV	0F	OF	16	Code : Symbole	0F	OF	16	Code : Symbole	0F	OF	16	Code : Symbole	0F	OF	16	Code : Symbole
10	10	17	DLE	10	10	17	ASCII	10	10	17	ASCII	10	10	17	ASCII	10	10	17	ASCII
11	11	18	DC1	11	11	18	EBCDIC	11	11	18	EBCDIC	11	11	18	EBCDIC	11	11	18	EBCDIC
12	12	19	DC2	12	12	19	OC COBOL	12	12	19	OC COBOL	12	12	19	OC COBOL	12	12	19	OC COBOL
13	13	20	DC3	13	13	20	Code : Symbole	13	13	20	Code : Symbole	13	13	20	Code : Symbole	13	13	20	Code : Symbole
14	3C	61	DC4	14	3C	61	ASCII	14	3C	61	ASCII	14	3C	61	ASCII	14	3C	61	ASCII
15	3D	62	NAK	15	3D	62	EBCDIC	15	3D	62	EBCDIC	15	3D	62	EBCDIC	15	3D	62	EBCDIC
16	32	51	SYN	16	32	51	OC COBOL	16	32	51	OC COBOL	16	32	51	OC COBOL	16	32	51	OC COBOL
17	26	39	ETB	17	26	39	Code : Symbole	17	26	39	Code : Symbole	17	26	39	Code : Symbole	17	26	39	Code : Symbole
18	18	25	CAN	18	18	25	ASCII	18	18	25	ASCII	18	18	25	ASCII	18	18	25	ASCII
19	19	26	EMV	19	19	26	EBCDIC	19	19	26	EBCDIC	19	19	26	EBCDIC	19	19	26	EBCDIC
1A	3F	64	SUB	1A	3F	64	OC COBOL	1A	3F	64	OC COBOL	1A	3F	64	OC COBOL	1A	3F	64	OC COBOL
1B	27	40	ESC	1B	27	40	Code : Symbole	1B	27	40	Code : Symbole	1B	27	40	Code : Symbole	1B	27	40	Code : Symbole
1C	1C	29	FSV	1C	1C	29	ASCII	1C	1C	29	ASCII	1C	1C	29	ASCII	1C	1C	29	ASCII
1D	1D	30	GSV	1D	1D	30	EBCDIC	1D	1D	30	EBCDIC	1D	1D	30	EBCDIC	1D	1D	30	EBCDIC
1E	1E	31	RSV	1E	1E	31	OC COBOL	1E	1E	31	OC COBOL	1E	1E	31	OC COBOL	1E	1E	31	OC COBOL
1F	1F	32	USV	1F	1F	32	Code : Symbole	1F	1F	32	Code : Symbole	1F	1F	32	Code : Symbole	1F	1F	32	Code : Symbole
20	40	65	▽	20	40	65	ASCII	20	40	65	ASCII	20	40	65	ASCII	20	40	65	ASCII
21	4F	80	!	21	4F	80	EBCDIC	21	4F	80	EBCDIC	21	4F	80	EBCDIC	21	4F	80	EBCDIC
22	7F	128	^	22	7F	128	OC COBOL	22	7F	128	OC COBOL	22	7F	128	OC COBOL	22	7F	128	OC COBOL
23	7B	124	#	23	7B	124	Code : Symbole	23	7B	124	Code : Symbole	23	7B	124	Code : Symbole	23	7B	124	Code : Symbole
24	5B	92	\$	24	5B	92	ASCII	24	5B	92	ASCII	24	5B	92	ASCII	24	5B	92	ASCII
25	6C	109	%	25	6C	109	EBCDIC	25	6C	109	EBCDIC	25	6C	109	EBCDIC	25	6C	109	EBCDIC
26	50	81	&	26	50	81	OC COBOL	26	50	81	OC COBOL	26	50	81	OC COBOL	26	50	81	OC COBOL
27	7B	126	,	27	7B	126	Code : Symbole	27	7B	126	Code : Symbole	27	7B	126	Code : Symbole	27	7B	126	Code : Symbole
28	4D	78	(	28	4D	78	ASCII	28	4D	78	ASCII	28	4D	78	ASCII	28	4D	78	ASCII
29	5D	94	)	29	5D	94	EBCDIC	29	5D	94	EBCDIC	29	5D	94	EBCDIC	29	5D	94	EBCDIC
2A	5C	93	•	2A	5C	93	OC COBOL	2A	5C	93	OC COBOL	2A	5C	93	OC COBOL	2A	5C	93	OC COBOL
2B	4E	79	+	2B	4E	79	Code : Symbole	2B	4E	79	Code : Symbole	2B	4E	79	Code : Symbole	2B	4E	79	Code : Symbole
2C	6B	108	,	2C	6B	108	ASCII	2C	6B	108	ASCII	2C	6B	108	ASCII	2C	6B	108	ASCII
2D	60	97	-	2D	60	97	EBCDIC	2D	60	97	EBCDIC	2D	60	97	EBCDIC	2D	60	97	EBCDIC
2E	4B	76	.	2E	4B	76	OC COBOL	2E	4B	76	OC COBOL	2E	4B	76	OC COBOL	2E	4B	76	OC COBOL
2F	61	98	/	2F	61	98	Code : Symbole	2F	61	98	Code : Symbole	2F	61	98	Code : Symbole	2F	61	98	Code : Symbole
30	F0	241	0	30	F0	241	ASCII	30	F0	241	ASCII	30	F0	241	ASCII	30	F0	241	ASCII
31	F1	242	1	31	F1	242	EBCDIC	31	F1	242	EBCDIC	31	F1	242	EBCDIC	31	F1	242	EBCDIC
32	F2	243	2	32	F2	243	OC COBOL	32	F2	243	OC COBOL	32	F2	243	OC COBOL	32	F2	243	OC COBOL
33	F3	244	3	33	F3	244	Code : Symbole	33	F3	244	Code : Symbole	33	F3	244	Code : Symbole	33	F3	244	Code : Symbole

## B.2 EBCDIC

EBCDIC				Indique l'absence de code de commande ou de symbole graphique	
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					
EBCDIC					
ASCII					
OC COBOL					
Code : Symbole					



# Glossaire

## **activité**

Partie d'un travail spécifiant le chargement et l'exécution d'un programme appelé "module chargeable".

## **administrateur système**

Responsable de la gestion technique d'un système d'exploitation.

## **//E-tape//**

Dérouleur de bande en cartouche au format 1/2 pouce à capacité améliorée (0,5 mm d'épaisseur, 330 mètres de long), également appelé long ou mince dérouleur de bande. Bicolore et utilisable uniquement sur des appareils à 36 pistes.

## **bande avec label standard**

Bande qui peut être lue avec label conforme aux normes ISO/ECMA (pour les dérouleurs de bande en cartouche, code EBCDIC uniquement).

## **bande non-standard**

Bande sans label standard

## **bande au format 1/2 pouce de longueur standard**

Bande en cartouche au format 1/2 pouce utilisée sur des dispositifs à 18 pistes (de 15 à 24 mètres de long). Également appelé bande courte ou épaisse. Peut être utilisé également sur des appareils à 36 pistes.

## **bibliothèque**

Fichier comportant des sous-fichiers appelés "unités de bibliothèque".

**bibliothèque de cartouche**

Module de stockage de cartouches équipé d'un automate qui réalise les opérations de montage et de démontage. Ces opérations sont toujours automatiques.

**bibliothèque CU**

Bibliothèque contenant les programmes compilés (unités compilées).

**bibliothèque d'entrée**

Bibliothèque où se trouvent les informations à lire.

**Bibliothèque de sortie**

Bibliothèque utilisée en lecture et en écriture.

**bibliothèque SL**

Bibliothèque origine, pouvant contenir n'importe quel type d'informations textuelles.

**bibliothèques de travail**

Bibliothèques d'entrée et de sortie utilisées pendant la session en cours.

**bloqué (travail)**

Etat dans lequel se trouve un travail temporairement suspendu par une commande HOLD.

**cartouche à 18 pistes**

Cartouche qui utilise le format d'enregistrement à 18 pistes. Ce type de cartouche est toujours au format 1/2 pouce de longueur standard.

**cartouche à 36 pistes**

Cartouche qui utilise le format d'enregistrement à 36 pistes. Ce type de cartouche peut être soit une bande au format 1/2 pouce de longueur standard, soit une //E-tape//.

**catalogue**

Fichier dans lequel sont enregistrés les liens entre un répertoire maître et ses fichiers.

**catalogue de site**

Catalogue contenant les informations relatives aux fichiers de site et à tous les autres répertoires, ainsi qu'à leur contenu.

**catalogue privé**

Catalogue contenant les informations relatives aux fichiers rattachés à un seul projet.

**catalogue système**

Catalogue contenant les informations relatives aux fichiers système (SYS.\*).

**chaîne protégée**

Chaîne de caractères entre apostrophes.

**charge maximum d'une classe (MCL)**

Nombre maximum de travaux pouvant se trouver simultanément suspendus ou en exécution dans une classe donnée.

**CI**

Voir "Intervalle de contrôle".

**classe d'appareils**

Type de périphérique (unité de disques, dérouleur de bandes classiques ou de bandes en cartouche).

**classe de travaux**

Groupe de travaux exécutant des tâches qui ont un rapport entre elles.

**compactage**

Méthode d'enregistrement sur cartouche, au standard IDRC (Improved Data Recording Capability), qui permet d'augmenter la capacité d'accueil de données sur cartouche.

**description de travail**

Ensemble d'ordres JCL fournissant des informations sur les travaux et commandant leur déroulement.

**désynchronisation des E/S (spooling)**

Stockage temporaire des entrées/sorties dans des fichiers intermédiaires pour traitement ultérieur, afin de ne pas bloquer le travail de l'utilisateur.

**dispositif à 18 pistes**

Dispositif qui utilise le format d'enregistrement à 18 pistes pour l'écriture des données. Il ne peut utiliser que des bandes au format 1/2 pouce de longueur standard.

**dispositif à 36 pistes**

Dispositif qui utilise le format d'enregistrement à 36 pistes pour l'écriture des données. Il peut utiliser des bandes au format 1/2 pouce de longueur standard et des //E-tapes//.

**éditeur des sorties**

Programme effectuant la sélection et l'impression des sorties en file d'attente pour le travail de service WRITER.

**Editeur ligne (TEXT EDITOR)**

Editeur de GCOS 7 servant à créer et modifier les unités de bibliothèque.

**EJR**

Voir ENTER\_JOB\_REQUEST.

**ENTER\_JOB\_REQUEST**

Commande opérateur demandant l'exécution d'un travail ou plus, ou la soumission d'un ensemble de travaux.

**en exécution (travail)**

Etat d'un travail faisant suite à la planification.

**en sortie (travail)**

Etat d'un travail dans lequel les sorties de celui-ci sont imprimées.

**état EX**

Voir "En exécution".

**état HOLD**

Voir "Bloqué"

**état IDLE**

Voir "Inactif".

**état IN**

Voir "Introduit"

**état OUT**

Voir "En sortie"

**état SCH**

Voir "En planification"

**état SUSP**

Voir "Suspendu"

**fichier**

Unité de stockage d'informations gérée par le système d'exploitation.

**fichier permanent**

Fichier créé en vue d'une utilisation à long terme, conservé dans un volume.

**fichier temporaire**

Fichier créé pour la durée d'une activité ou d'une session (supprimé en fin d'activité ou de session).

**fichier UFAS séquentiel indexé**

Fichier disque contenant des index qui servent à localiser les articles.

## **FIFO**

Abréviation de "First in, first out" (premier entré, premier sorti).

Ordre d'exécution des travaux selon lequel le premier travail entré est traité en premier par le système.

## **GCL**

Abréviation de "GCOS Command Language".

Ensemble de commandes à la disposition des utilisateurs IOF pour exécuter les fonctions GCOS 7 sur DPS 7000.

## **GCOS 7**

Abréviation de "General Comprehensive Operating System".

Système d'exploitation du DPS 7000.

## **HJ**

Voir HOLD\_JOB.

## **HOLD\_JOB**

Commande opérateur servant à placer un travail à l'état HOLD (bloqué).

## **inactif (travail)**

Etat des travaux de service JTRA et WRITER lorsqu'ils ne s'exécutent pas.

## **intervalle de contrôle**

Unité de transfert entre la mémoire principale et un fichier disque.

## **introduit (travail)**

Premier état d'un travail correspondant à la traduction du JCL dans un format interne propre à l'exécution.

## **IOF**

Abréviation de "Interactive Operation Facility".

Logiciel d'exploitation en temps partagé de GCOS 7.

**JCL**

Abréviation de "Job Control Language".

Langage permettant de gérer l'exécution des travaux sous GCOS 7.

**JOB\_OUT**

Historique contenant les informations système sur l'exécution de toute activité générant une sortie.

**JOR**

Abréviation de "Job Ocurrence Report".

Historique contenant les informations système sur l'exécution de tout travail générant une sortie.

**JTRA**

Voir "Traducteur JCL".

**lecteur des entrées**

Dans un travail donné, module chargeable qui lit les entrées et affecte l'appareil d'entrée.

**lecteur de travaux**

Le lecteur de travaux lit un train de descriptions de travaux, analyse chacune d'entre elles, traduit le JCL en format exploitable par GCOS, et stocke séparément ordres JCL et données en vue de la planification et de l'exécution des travaux.

**mécanisme SYSIN**

Permet aux programmes lisant des données à partir d'un appareil d'entrée de s'exécuter concurremment sans problèmes d'affectation d'appareils.

**média**

disque, bande ou bande en cartouche.

**module chargeable**

Programme connu du système au moment de son exécution.

**multiprogrammation**

Environnement dans lequel deux ou plusieurs travaux s'exécutent concurremment.

**nom d'accès complet**

Nom complet d'un fichier, comportant le ou les noms de répertoires nécessaires pour y accéder.

**nom simple**

Chacune des composantes du nom d'accès complet d'un fichier.

**noms symboliques**

Ensemble de noms spéciaux (LIB, INLIB1, INLIB2 et INLIB3) désignant les bibliothèques de travail.

**numéro de passage (RON)**

Numéro affecté à un travail après sa soumission au système.

**opérateur principal**

Opérateur gérant le fonctionnement quotidien d'un système GCOS 7.

**ordre JCL de base**

Ordre JCL déclenchant une opération système spécifique.

**ordre JCL étendu**

Ordre JCL développé ultérieurement en une série d'ordres JCL de base, et équivalant à une description d'activité complète.

**paramètre booléen**

Paramètre dont la valeur est 0 (NON) ou 1 (OUI).

**paramètre obligatoire**

Paramètre dont la spécification est indispensable à l'exécution de l'ordre.

**passage d'un travail**

Cycle de vie complet d'un travail.

**planificateur des travaux**

Gère la charge du système au niveau travail et permet à l'utilisateur d'organiser le travail.

**priorité de planification**

Ordre dans lequel les travaux sont placés en file d'attente en vue de leur exécution.

**priorité de sortie**

Ordre dans lequel les travaux sont imprimés par WRITER.

**processeur**

Groupe de commandes correspondant à des fonctions spécifiques de GCOS 7.

**projet**

Groupe logique d'utilisateurs.

**racine**

Point de départ d'une structure arborescente.

**RELEASE\_JOB**

Commande opérateur faisant passer un travail de l'état HOLD à l'état SCH.

**répertoire**

Groupe de fichiers dans une structure hiérarchique.

**répertoire de travail**

Répertoire contenant les fichiers de travail de l'utilisateur.

**répertoire maître**

Répertoire de premier niveau dans une structure arborescente.

**RJ**

Voir RELEASE\_JOB.

**RON**

Voir "Numéro de passage".

**rubrique de description d'activité**

Partie d'une rubrique de description de travail. Une rubrique de description de travail peut comporter une ou plusieurs rubriques de description d'activité.

**rubrique de description d'entrées**

Partie d'un travail introduite par un ordre JCL \$INPUT et close par un ordre JCL \$ENDINPUT. Une rubrique de description d'entrées contient les articles de données à associer à une ou plusieurs activités d'un travail.

**rubrique de description de données**

Partie d'un travail introduite par un ordre JCL \$DATA et close par un ordre JCL \$ENDDATA. Une rubrique de description de données contient les données à charger dans une bibliothèque spécifique.

**rubrique de description de travail**

Ensemble des ordres JCL formant une description de travail. Une rubrique de description de travail commence par un ordre \$JOB et se termine par un ordre \$ENDJOB.

**spooling**

Formé à partir de "Simultaneous Peripheral Operations On Line".  
Voir "Désynchronisation des E/S".

**statut d'un fichier**

Caractère catalogué, non catalogué ou temporaire d'un fichier.

**suspendu (travail)**

Etat d'un travail dont l'exécution est momentanément interrompue.

**SYS.OUT**

Fichier système standard contenant les sorties de travaux à imprimer.

**traducteur JCL**

Travail de service dont le rôle est de traduire en format interne le JCL d'un travail qui vient d'être soumis.

**train d'entrées**

Travail constitué d'une ou plusieurs descriptions de travaux soumis au système.

**travail traitement par lots**

Travail exécuté sans intervention de l'opérateur.

**travail traitement par lots conventionnel**

Travail exécutant des séquences d'ordres JCL.

**unité de bibliothèque**

Sous-fichier d'un fichier bibliothèque.

**utilitaire de maintenance des bibliothèques**

Utilitaire offrant à l'utilisateur diverses commandes permettant de gérer et, éventuellement, modifier des bibliothèques.

**volume**

Disque, bande ou bande en cartouche contenant les informations.

**volume non résidant**

Volume chargé uniquement lorsqu'il est nécessaire.

**volume résidant**

Volume à la disposition du système (en ligne) de façon permanente.

**WRITER**

Travail de service gérant l'impression des sorties de travaux.

**zone de données**

Unité d'information à l'intérieur d'un article.



# Index

## CARACTERES SPECIAUX

#	Voir "Dièse"
\$	Voir "Dollar"
'	Voir "Apostrophes"
-	Voir "Trait d'union"
.	Voir "Point"
=	Voir "Signe égal"
;	Voir "Point-virgule"
:	Voir "Deux-points"
 \$	
\$, voir signe dollar	2-1
\$DATA	2-1, 4-32
BILLING	4-34
CKSTAT	4-36
CONTCHAR	4-36
description-bibliothèque-de sortie	4-34
END	4-35
ENDCHAR	4-35
FORMAT	4-35
nom-unité-bibliothèque	4-34
PRINT	4-36
PROJECT	4-34
RECSIZE	4-36
REPLACE	4-36
TYPE	4-35
USER	4-34
\$ENDDATA	1-3, 2-1, 4-53
\$ENDINPUT	1-3, 2-1, 4-54
nom-rubrique-de-description-d'entrées	4-54
\$ENDJOB	1-1, 2-1, 4-55
\$IN	
PUTCKSTAT	4-71
\$INPUT	1-3, 2-1, 4-62
chaîne8	4-67
CKSTAT	4-65

COBOL	4-66
CONTCHAR	4-65, 4-69
CVVALUES	4-71
DATA	4-66
DATASSF	4-66
DOLLAR	4-67
END	4-67
ENDCHAR	4-65, 4-68
ENDINPUT	4-67
EOF	4-67
FORMAT	4-65, 4-67
INFILE	4-71
JVALUES	4-71
MATCH	4-67
MEDIA=*	4-71
MEDIA=nom-volume	4-71
nom-rubrique-de-description-d'entrées	
	4-66
paramétrage	4-65
PRINT	4-70
TYPE	4-66
\$JOB	1-1, 2-1, 4-77
ABORT	4-81
ALL	4-81
BILLING	4-79
CLASS	4-81
HOLD	4-81
HOLDOUT	4-82
HOST	4-82
JOBLANG	4-83
JOR	4-81
LIST	4-80
NO	4-81
nom-travail	4-78
NORMAL	4-81
NSTARTUP	4-80
PRIORITY	4-82
PROJECT	4-79
RECSIZE	4-82
REPEAT	4-82
SOURCE	4-80
USER	4-78

\$SWINPUT	2-1, 4-131	nom-rubrique-de-description-d'entrées	4-13
CONSOLE	4-132	NORMAL	4-20
description-bibliothèque-d'entrée	4-132	NVOLWR	4-17
description-fichier-d'entrée-séquentiel	4-132	ONEWRITE	4-20
		OPTIONAL	4-17
		PASS	4-21
		POOL	4-25
		READ	4-21
		RECOVERY	4-21
		RESIDENT	4-18
ALLOCATE	4-3	SHARE	4-20
CHECK	4-5	SPREAD	4-21
INCRSIZE	4-5	SPWRITE	4-21
nom-fichier-interne	4-5	SUBFILE	4-20
SIZE	4-5	TEMPRY	4-14
UNIT	4-5	UNCAT	4-14
apostrophes	2-7, 3-8	UNLOAD	4-23
appareil		VOLSET	4-19
classe	3-11	WORK	4-18
identification	3-11	WRITE	4-21
nom	3-11		
ASSIGN	4-6		
*	4-14, 4-18		
ABEND	4-24	chaîne protégée	2-7
ACCESS	4-21	charge maximum	
ALLREAD	4-21	d'une classe	1-15
CAT	4-14	du système	1-12
CATALOG	4-15	classe d'appareils	
CATNOW	4-15	dérouleur de bandes	3-15
concaténation de fichiers	4-10	dérouleur de bandes en cartouche	3-13
DEASSIGN	4-22	imprimante	3-15
DEFER	4-17	unité de disque	3-14
DENSITY	4-28	Classe de travaux	
description-classe-d'appareils	4-18	attributs	1-14
DEVCLASS	4-18	charge d'une ~	1-14
DIR	4-20	classe de travaux	1-13
DUMMY	4-14	COMMENT	4-30
DVIDLIST	4-18	chaîne-commentaire	4-30
END	4-21	conservation d'une activité à une autre	1-27
EXPDATE	4-16	CONSOLE	4-31
FILESTAT	4-14	nom-utilisateur	4-31
FIRST	4-25	CONTINUE	2-3
FIRSTVOL	4-26	conversion des caractères	7-1
FREE	4-20		
FSN	4-27		
LABEL	4-26		
LEAVE	4-22		
liste-identificateurs-d'appareils	4-18		
MEDIA	4-18		
MONITOR	4-20		
MOUNT	4-25	DEFINE	4-38
NBEFN	4-28	ABORT	4-44
NEXT	4-25	AFTER	4-42
nom-fichier-externe	4-13	ANSI	4-48
nom-fichier-interne	4-13	BEFORE	4-42
nom-fichier-temporaire	4-13	BLKOFF	4-49
		BLKSIZE	4-48

**A**

activité	1-3
ALLOCATE	4-3
CHECK	4-5
INCRSIZE	4-5
nom-fichier-interne	4-5
SIZE	4-5
UNIT	4-5
apostrophes	2-7, 3-8
appareil	
classe	3-11
identification	3-11
nom	3-11
ASSIGN	4-6
*	4-14, 4-18
ABEND	4-24
ACCESS	4-21
ALLREAD	4-21
CAT	4-14
CATALOG	4-15
CATNOW	4-15
concaténation de fichiers	4-10
DEASSIGN	4-22
DEFER	4-17
DENSITY	4-28
description-classe-d'appareils	4-18
DEVCLASS	4-18
DIR	4-20
DUMMY	4-14
DVIDLIST	4-18
END	4-21
EXPDATE	4-16
FILESTAT	4-14
FIRST	4-25
FIRSTVOL	4-26
FREE	4-20
FSN	4-27
LABEL	4-26
LEAVE	4-22
liste-identificateurs-d'appareils	4-18
MEDIA	4-18
MONITOR	4-20
MOUNT	4-25
NBEFN	4-28
NEXT	4-25
nom-fichier-externe	4-13
nom-fichier-interne	4-13
nom-fichier-temporaire	4-13

**C**

chaîne protégée	2-7
charge maximum	
d'une classe	1-15
du système	1-12
classe d'appareils	
dérouleur de bandes	3-15
dérouleur de bandes en cartouche	3-13
imprimante	3-15
unité de disque	3-14
Classe de travaux	
attributs	1-14
charge d'une ~	1-14
classe de travaux	1-13
COMMENT	4-30
chaîne-commentaire	4-30
conservation d'une activité à une autre	1-27
CONSOLE	4-31
nom-utilisateur	4-31
CONTINUE	2-3
conversion des caractères	7-1

**D**

DEFINE	4-38
ABORT	4-44
AFTER	4-42
ANSI	4-48
BEFORE	4-42
BLKOFF	4-49
BLKSIZE	4-48

BOTH	4-42	SKIP	4-44
BPB	4-43	SLEW	4-50
BPIOC	4-45	STAT	4-47
BSN	4-49	SYSOUT	4-44
BUFPOLL	4-47	WRCHECK	4-44
CHi	4-51	description de travail	1-1
chiffres8	4-43	description-bibliothèque-d'entrée	3-22
CIFSP	4-46	description-bibliothèque-d'impression	3-23
CISIZE	4-46	description-bibliothèque-de-sortie	3-23
CKPTLIM	4-43	description-fichier-d'entrée	3-20
COLLATE	4-49	description-fichier-d'entrée-séquentiel	3-19
COMPACT	4-46	description-fichier-d'impression	3-21
CONVERT	4-49	description-fichier-de-sortie	3-21
DATACODE	4-49	description-fichier-de-sortie-séquentiel	3-20
EOF	4-52	description-fichier-de-travail	3-21
EOV	4-43	désynchronisation des E/S	1-22
ERROPT	4-44	deux-points	2-3
EXCL	4-47	DEVCLASS	3-11
FF1	4-51	dièse (#)	2-4, 2-5, 2-7
FF2	4-51	dollar (\$)	2-1
FILEFORM	4-48	DVIDLIST	3-11
FILEORG	4-46		
FORMHT	4-50		
FPARAM	4-46		
FRIOC	4-45		
FUNCMASK	4-49		
HOF	4-51	éditeur des sorties	1-7
IGNORE	4-44	éléments d'un ordre JCL	2-3
JOURNAL	4-42	ENDSTEP	1-3, 4-56
KEYLOC	4-45	espace	2-1, 2-4
KEYSIZE	4-46	état bloqué, HOLD	1-6
LOCKMARK	4-48	état en exécution, EX	1-6
LTRKSIZE	4-46	état en sortie, OUT	1-6
MARGIN	4-50	état EX	1-7
MSG	4-52	état HOLD	1-7
NBBUF	4-43	état IN	1-7
NBSN	4-49	état OUT	1-7
NCOMPACT	4-46	état SCH	1-7
NCONVERT	4-49	état SUSP	1-7
NMSG	4-52	état suspendu, SUSP	1-6
NO	4-43	étiquette	2-1, 2-3
nom-fichier-interne	4-42	EXDIR	4-57
NONE	4-42	EXECUTE	1-5, 1-26, 4-58
NORMAL	4-47	description-fichier-d'entrée-séquentiel	
NPROMPT	4-52		4-60
NSLEW	4-50	PRTDEF	4-60
NSTD	4-48	PRTFILE	4-60
NSYSOUT	4-44	PRTOUT	4-60
NTRNCSSF	4-45	SIZEOPT	4-61
OPTIMIZE	4-44	STEPOPT	4-61
PRDEN	4-50	VALUES	4-60
PROMPT	4-51	exécution d'une activité	1-17
RAHEAD	4-45		
READLOCK	4-47		
RECFORM	4-48		
RECSIZE	4-48		
RETCODE	4-44		

**E**

**F**

Fichier	
bibliothèque	3-7
fichier	
bande ANS	3-6
bibliothèque	3-2
catalogué	3-2
disquette	3-6
non catalogué	3-2, 3-6
permanent	3-3
statut	3-9
SYS.IN	3-2
temporaire	3-2, 3-7
Fichier SYSOUT	
permanent	4-135
standard	4-135
fichier SYSPRINT	
permanent	1-24
standard	1-24
file d'attente du planificateur de travaux	
	1-12, 1-14
FILESTAT	3-9
fin d'activité	1-17

**G**

groupe de paramètres standard	3-17
-------------------------------	------

**H**

HJ	
voir HOLD_JOB:	1-7
HOLD_JOB	1-7

**I**

Interpréteur d'ordres JCL	1-17
INVOKER	4-73
description-bibliothèque-d'entrée	4-74
LIST=ALL	4-76
nom-rubrique-de description-d'entrées	
1	4-74
nom-unité-bibliothèque	4-74
SYS	4-75
UPDATE	4-76
VALUES	4-75
IOF	1-14
IOF, voir Interactive Operation Facility	1-6

**J**

JCL	
de base	1-5
enregistré	1-26
étendu	1-5
JOR	1-24
JOR, voir Job Occurrence Report	1-11
Journal	
Après	1-28
Avant	1-28
JTRA	1-9, 1-11
JUMP	1-18, 4-84
\$ENDJOB	4-88
CONTINUE	4-85
IOF	4-87
JUMP	4-88
nom-étiquette	4-85
SEV	4-86
STATUS	4-86
STEP*	4-88
SWi	4-86

**L**

LASTVOL	4-26
lecteur de travaux	1-7, 1-9
lecteur des entrées	1-9, 1-11, 1-23, 2-1
LET	4-91
hexa8	4-92
SEV	4-92
SW	4-92
SWi	4-92
LIB	4-93
BIN	4-95
CU	4-95
description-bibliothèque-d'entrée	4-95
LM	4-95
SL	4-95

**M**

MAXLOAD, voir charge maximum	1-15
MDJ	
voir MODIFY_JOB	1-14
MDD, voir commande MODIFY_LOAD	1-14
MESSAGE	4-97
MODIFY_JOB	1-14
module chargeable	1-3
MODVL	4-98
valeur-mot-clé-i	4-99
valeur-paramètre-i	4-99

**N**

nom d'appareil	3-16
nom d'ordre	2-1, 2-3
nom de fichier	3-3
élément	3-3
externe	3-3
non standard	3-8
nom de rubrique de description d'entrées	3-7
noms d'ordres JCL	6-1
noms de volumes	
non conformes	3-10

**O**

ordre JCL	
éléments	2-3
ordres JCL étendus	4-4, 4-31, 4-88, 4-93
OUTVAL	4-101

**P**

paramètre d'implantation	3-1, 3-2
paramètres à mot-clé	2-6
paramètres autodéfinis	2-6
paramètres positionnels	2-4
paramètres-define	3-24
paramètres-size	3-24
paramètres-step	3-24
paramètres-sysout	3-24
Partage des fichiers	1-27
passage d'un travail	1-6
planificateur de travaux	1-12
point-virgule	2-1, 2-3
POOL	4-103
classe-d'appareils	4-104
MAX	4-105
nom-d'appareil	4-104
PREFIX	4-106
préfixage automatique	3-6
priorité	
d'exécution	1-15
de planification	1-12, 1-14

**Q**

QASSIGN	4-108
ACCESS	4-110
IN	4-110
INOUT	4-110
LIN	4-110
nom-file-d'attente-externe1	4-109

nom-file-d'attente-symbolique	4-109
nom-sous-file-d'attente-symbolique	4-110
OUT	4-110
REPLY	4-111
RR	4-110
SITE	4-111

**R**

RBF	1-6
références aux paramètres	1-26, 2-2
relance	1-28
RELEASE	4-112
nom-travail	4-113
PASS	4-113
SWITCHES	4-113
SWITCHi	4-113
RELEASE_JOB	1-7, 4-81
RELEASE_OUTPUT	4-82
répertoire maître	3-3
REPORT	4-114
RJ	
voir RELEASE_JOB	1-7
RO	
voir RELEASE_OUTPUT	4-118, 4-139
RON	4-9, 4-78, 4-79, 4-150
RON, voir Run Occurrence Number	1-6
routine de lancement d'activité	1-17
rubrique de description d'activité	1-3
rubrique de description d'entrées	1-1, 1-3
rubrique de description de données	1-1, 1-3
rubrique de description de travail	1-1
RUN	1-5, 1-26, 1-27, 4-115
CLASS	4-118
DELETE	4-119
HOLD	4-118
HOLDOUT	4-118
INDEF	4-117
INDEFi	4-117
INFILEi	4-117
JOBS	4-118
PASS	4-119
PRIORITY	4-118
SIZEOPT	4-120
STEPOPT	4-120
SWITCHES	4-119
SWITCHi	4-119
VALUES	4-119

**S**

SEND	4-121
chaîne105	4-122
MAIN	4-122
nom-utilisateur	4-122
signe égal (=)	2-6
SIZE	4-123
CHPPAGE	4-124
espace-d'exécution-déclaré	4-124
NBBUF	4-125
POOLSIZE	4-125
SORTWORK	1-5
station DJP	1-6
STEP	1-3, 4-126
CPTIME	4-128
DEBUG	4-129
description-bibliothèque-d'entrée	4-127
DUMP	4-128
ELAPTIME	4-128
LINES	4-128
MAXMEM	4-129
MINMEM	4-129
nom-module-chargeable	4-127
OPTIONS	4-129
REPEAT	4-129
XPRTY	4-128
syntaxe des noms	3-18
SYS.IN	1-9, 1-23
SYS.OUT	1-24
SYSIN	
fichier système	1-9
mécanisme	1-9, 1-23
SYSOUT	
BANINF	4-134
BANLEVEL	4-139
BANNER	4-140
chiffres5	4-139
CLASS	4-137
COPIES	4-140
DEFER	4-138
DELETE	4-141
DEST	4-141
DEVCLASS	4-140
format	1-25
HOLD	4-139
IMMED	4-138
JOB	4-138
mécanisme	1-25
MEDIA	4-140
NAME	4-139
NBANNER	4-139
NDELETE	4-141
NHOLD	4-139
nom-fichier-interne	4-137

NSLEW	4-141
PRIORITY	4-137
STEP	4-138
WHEN	4-138

**T**

traducteur JCL	1-9, 1-11
train d'entrées	1-1, 1-10
trait d'union	2-2, 2-8
travaux de service	1-14, 1-20
type de fichier	3-1

**V**

valeur des paramètres	3-17
VALUES	4-143
valeur-mot-cléi	4-144
valeur-paramètreti	4-144
virgule (,)	2-1, 2-4, 2-5, 2-7
volume	
identification	3-9
nom	3-10
non résident	3-10
résident	3-9

**W**

WRITER	4-146
ALL	4-151
ASA	4-151
COPIES	4-150
DATAFORM	4-151
description-bibliothèque-d'entrée	4-149
description-fichier-d'entrée séquentiel	4-148
DOF	4-151
FPARAM	4-152
IMMED	4-150
JOB	4-150
NFPARAM	4-152
NUMBER	4-152
PART	4-149
PRINTER	4-152
REPORT	4-151
SARF	4-151
SELECT	4-150
SSF	4-151
SUBFILES	4-149
WHEN	4-150