



VisualAge Pacbase 2.5

**LANGAGE STRUCTURE
MANUEL DE REFERENCE**

DDSTR000251F

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section "Remarques" de la page suivante.

En application de votre contrat de licence, vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir du site Web du Support Technique :

<http://www.software.ibm.com/ad/vapacbase/support.htm>

La section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière édition disponible du présent document.

Première Edition (Avril 1998)

La présente édition s'applique à :

- VisualAge Pacbase Version 2.5

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante :

<http://www.software.ibm.com/ad/vapacbase/support.htm>

ou en nous adressant un courrier à :

IBM Paris Laboratory
Support VisualAge Pacbase
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

IBM pourra disposer comme elle l'entendra des informations contenues dans vos commentaires, sans aucune obligation de sa part.

© Copyright International Business Machines Corporation 1983, 1999. Tous droits réservés.

REMARQUES

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente.

Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM.

Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à :

IBM Paris Laboratory
Département SMC
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

MARQUES

IBM est une marque d'International Business Machines Corporation, Inc.
AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, PACBASE, RACF, RS/6000, SQL/DS, TeamConnection et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

TABLE DES MATIERES

| | |
|---|------------|
| 1. INTRODUCTION | 7 |
| 1.1. VISUALAGE PACBASE..... | 8 |
| 1.2. OBJET DU MANUEL | 11 |
| 1.3. LANGAGE STRUCTURE : PRESENTATION | 12 |
| 1.4. PRESENTATION GENERALE..... | 13 |
| 1.5. ENTITE ET LIGNES GERES | 15 |
| 2. PROGRAMMES | 17 |
| 2.1. PROGRAMMES : PRESENTATION..... | 18 |
| 2.2. STRUCTURE D'UN PROGRAMME | 19 |
| 2.3. DEFINITION D'UN PROGRAMME..... | 21 |
| 3. MACRO-STRUCTURES..... | 30 |
| 3.1. MACRO-STRUCTURES : PRESENTATION | 31 |
| 3.2. APPEL DE MACRO-STRUCTURES (-CP)..... | 35 |
| 3.3. MACRO-STRUCTURES NON VENTILEES | 42 |
| 4. EMPLOI DES STRUCTURES DE DONNEES | 44 |
| 4.1. APPEL DES STRUCTURES DE DONNEES (-CD)..... | 45 |
| 4.2. MODIFICATION DU DEBUT DE PROGRAMME (-B)..... | 69 |
| 4.3. ZONES DE TRAVAIL ET DE LIEN (-W)..... | 73 |
| 5. LANGAGE STRUCTURE..... | 80 |
| 5.1. DESCRIPTION DES TRAITEMENTS (-P)..... | 81 |
| 6. TITRES, CONDITIONS ET NIVEAUX DES TRAITEMENTS | 108 |
| 6.1. TITRES ET CONDITIONS DES TRAITEMENTS (-TC) | 109 |
| 6.2. TITRES ET NIVEAUX DES TRAITEMENTS (-TO) | 120 |
| 7. REPERTOIRE D'ACCES | 123 |
| 7.1. ACCES AUX ECRANS | 124 |
| 7.2. OPTIONS D'AFFICHAGE EN T.P..... | 128 |
| 7.3. CODES ACTION EN T.P. | 129 |
| 7.4. MODE D'ACCES EN BATCH | 130 |
| 7.5. EDITION ET GENERATION | 136 |
| 8. DESCRIPTION D'UN PROGRAMME GENERE..... | 137 |
| 8.1. INTRODUCTION | 138 |
| 8.2. IDENTIFICATION DIVISION..... | 139 |
| 8.3. ENVIRONNEMENT DIVISION | 140 |
| 8.4. DATA DIVISION FILE SECTION | 142 |
| 8.5. DEBUT DE WORKING-STORAGE SECTION | 146 |
| 8.6. VARIABLES ET INDICES | 149 |
| 8.7. ZONES DE TRAVAIL ET DE LIEN | 152 |
| 9. ANNEXES | 156 |
| 9.1. SOURCE COBOL (LIGNES '9') | 157 |

1. INTRODUCTION

1.1. VISUALAGE PACBASE

L'OFFRE VisualAge Pacbase

VisualAge Pacbase est un AGL fonctionnant sous des serveurs mainframe, OS/2, UNIX ou Windows NT. VisualAge Pacbase a pour but d'assurer la gestion complète d'applications informatiques.

La cohérence de l'ensemble est assurée, car toutes les informations sont regroupées dans une Base de Spécifications unique et sont gérées de manière unique.

Les Produits de l'offre VisualAge Pacbase

L'offre VisualAge Pacbase est une offre modulaire qui s'articule autour de deux produits principaux - Pacdesign pour la Conception d'applications et Pacbench pour le Développement.

Pacdesign et Pacbench permettent d'alimenter la Base de Spécifications et de couvrir le cycle de vie des applications; un certain nombre de modules sont disponibles pour chacun de ces deux produits.

Modules de base

Dictionnaire
Langage Structuré
Gestionnaire de Documentation Personnalisée (GDP-GDP+)

Générateurs

Dialogue
Pacbench Client/Serveur
Langage Batch
COB / Generator

Support Bases de Données

DBD
DBD-SQL

Habillage d'applications

Pacbench Automatic Windowing (PAW) (versions antérieures à VisualAge Pacbase 2.0)
Pacbase Web Connection

Contrôle qualité

Pacbench Quality Control (PQC)
Extension : Quality Control Extensibility

Gestionnaire de Tables

Pactables

Mise en production et suivi

Environnement de production (PEI)

PacTransfer

Development Support Management System (DSMS)

Module micro : DSMS habillé (pour les versions antérieures à VisualAge

Pacbase 2.0)

Services complémentaires

Pac/Impact

Personnalisation (Extension du Dictionnaire)

Pacbase Access Facility (PAF-PAF+)

DSMS Access Facility (DAF)

Méthodes (Merise, YSM, etc.)

Gestionnaire de Bases d'Informations Réparties (GBIR)

Normalisation d'Entités (RMEN)

Analyse d'Activité (ACTI)

Interface Sécurité RACF/TOPSECRET

ENDEVOR

Pont VisualAge Smalltalk-VisualAge Pacbase

Pont Team Connection-VisualAge Pacbase

| | | |
|-----------------|------|----|
| INTRODUCTION | PAGE | 11 |
| OBJET DU MANUEL | | 1 |
| | | 2 |

1.2. OBJET DU MANUEL

OBJET DU MANUEL

Ce manuel décrit toutes les spécifications du module Langage Structuré utilisables sans les modules complémentaires Dialogue et Langage Batch.

Il décrit en outre un certain nombre de spécifications liées à ces modules afin d'éviter une nouvelle description des sous-entités concernées dans les manuels correspondants.

CONNAISSANCES PREALABLES

L'utilisateur doit avoir suivi une session de formation aux modules DICTIONNAIRE et LANGAGE STRUCTURE.

Il doit avoir pris connaissance :

- . du Manuel de Référence DICTIONNAIRE,
- . du Manuel d'UTILISATION.

1.3. LANGAGE STRUCTURE : PRESENTATION

PRINCIPES DE DESCRIPTION

La description des entités et des lignes gérées par VisualAge Pacbase est en général composée de deux parties :

- . Une introduction expliquant le but et les caractéristiques générales de l'entité ou de la ligne.
- . Une description exhaustive, pour chaque ligne, des zones d'entrée dans les écrans ou dans les bordereaux de saisie batch.

Dans la mesure où les écrans et les bordereaux de saisie contiennent les mêmes zones, leur description est commune.

Dans la description, chaque zone est repérée par un numéro d'ordre correspondant le plus souvent à la saisie sur écran.

Les numéros d'ordre sont repris dans les maquettes d'écrans présentées dans les différents manuels de référence ainsi que sur les bordereaux de saisie batch correspondants.

Pour certaines descriptions, il est possible de trouver de légères différences entre les zones à saisir à l'écran et celles figurant sur les bordereaux batch. Elles s'expliquent par le manque de souplesse du batch par rapport au TP, qui entraîne souvent la saisie de zones supplémentaires en batch (par exemple certains indicatifs qui figurent déjà sur l'écran correspondant).

D'autre part, l'ordre des zones peut être différent pour les écrans et les bordereaux batch. Dans ce cas, les numéros repérant les zones ne sont pas en séquence croissante (soit sur la maquette de l'écran, soit sur le bordereau batch).

>>>> Si vous utilisez la Station de Travail VisualAge Pacbase, l'interface graphique des fenêtres correspondantes est documentée dans le Manuel de Référence de la Station de Travail.

1.4. PRESENTATION GENERALE

LANGAGE STRUCTURE : PRESENTATION

Le LANGAGE STRUCTURE est un langage de haut niveau qui permet aux équipes de programmation de réaliser tout programme de gestion. Ce langage est obligatoirement lié avec le DICTIONNAIRE, il offre les avantages suivants :

- . Une plus grande concision dans l'écriture que le COBOL par la simplification des ordres et de la ponctuation,
- . Une hiérarchisation des traitements inexistante en COBOL,
- . Une structuration des traitements selon les règles habituelles de la programmation structurée, par l'emploi d'opérateurs (BLOCK, IF THEN, ELSE, DO WHILE, DO UNTIL, CASE OF, ...), qui n'existent pas en COBOL,
- . Une portabilité améliorée, par une conception indépendante de tout constructeur et une génération de programmes COBOL adaptée à chaque type de matériel.

DESCRIPTION

Les possibilités offertes par ce module sont :

- . La définition des Programmes, par l'intermédiaire d'un écran qui contiendra ses caractéristiques générales (nom en clair, type, mots-clés explicites, etc.),
- . L'appel des Structures de Données, qui sont déjà décrites dans le DICTIONNAIRE, qu'on pourra utiliser autant de fois que nécessaire dans les Programmes,
- . La définition des zones de travail, soit par l'appel de Structures de Données existantes, soit par la description de Structures de Données propres à chaque Programme,
- . La définition de Macro-Structures, c'est-à-dire d'ensembles de lignes de LANGAGE STRUCTURE éventuellement paramétrées,
- . L'appel de ces Macro-Structures dans les Programmes, avec possibilité d'appeler plusieurs fois une même Macro-Structure Paramétrée dans un même Programme,
- . La description des traitements spécifiques au Programme, par un langage de haut niveau.

REFERENCES CROISEES

L'utilisation du module DICTIONNAIRE permet d'obtenir des références croisées indiquant l'utilisation des Rubriques et des Structures de Données dans les Programmes écrits à l'aide du module LANGAGE STRUCTURE.

Ces références croisées, particulièrement utiles lors de la phase de réalisation, constituent un outil essentiel en phase de maintenance, car elles permettent d'estimer immédiatement l'impact des modifications.

TYPES DE PROGRAMMES REALISABLES

Le LANGAGE STRUCTURE permet d'écrire des Programmes de tous types, batch ou transactionnels, en utilisant les possibilités offertes par les Macro-Structures Paramétrées (M.S.P.).

En effet, les M.S.P. peuvent effectuer les traitements technologiques associés aux Ecrans, ou à un système de gestion de base de données particulier, et être appelées dans tous les Programmes. Dans le cas de Programmes transactionnels, la description des Ecrans n'est alors pas assurée automatiquement par le module LANGAGE STRUCTURE.

1.5. ENTITE ET LIGNES GEREEES

ENTITE ET LIGNES GEREEES

Le module LANGAGE STRUCTURE gère une seule entité : le PROGRAMME, défini et décrit par les lignes suivantes :

- . définition de Programme,
- . appel de Macro-structures,
- . appel de Structures de Données,
- . modifications du début de Programme,
- . description des zones de travail et de lien,
- . description des traitements.

La ligne de définition du Programme (TP : 'P', batch : '0' <zéro>) permet de définir son code, son nom en clair et ses caractéristiques principales.

Les lignes d'appel de Macro-structures (TP : '-CP', batch : 'M') permettent d'inclure dans un Programme des éléments décrits dans d'autres entités Programme en remplaçant les paramètres indiqués dans celles-ci par des valeurs particulières.

Les lignes d'appel de Structures de Données (TP : '-CD', '-HCD', batch : '1') permettent d'obtenir la description des zones d'entrée-sortie du Programme.

Les lignes de modification de début de Programme (TP : '-B', batch : 'D') permettent de modifier les descriptions obtenues à partir des appels précédents.

Les lignes de description de zones de travail et de lien (TP : '-W', batch : '7') permettent d'appeler des Structures de Données ou des Rubriques et de compléter les descriptions obtenues.

Les lignes de description de traitements (TP : '-P', batch : 'P') permettent d'écrire sous une forme portable, structurée et hiérarchisée des séquences d'instructions.

Comme toutes les entités PACBASE, les Programmes peuvent être documentés par des lignes de Documentation Généralisée et par la ventilation de Textes (Cf. Manuel de Référence DICTIONNAIRE).

PROGRAMMES ISSUS DE REVERSE ENGINEERING

Les programmes issus du module REVERSE ENGINEERING ne comportent que des zones de travail (-W) et de la procédure en source COBOL (-SC).

Il est possible de les compléter par du langage structuré (-W et -P), et par des appels de macro-structures (-CP), et de les régénérer. Les lignes -CD et -B sont ignorées.

Pour la mise en oeuvre de ces fonctions, se reporter au manuel GENERATEUR COBOL.

2. PROGRAMMES

2.1. PROGRAMMES : PRESENTATION

PROGRAMMES : PRESENTATION

L'entité PROGRAMME a pour objet la réalisation de tous les traitements de type batch définis en phase d'analyse détaillée.

CARACTERISTIQUES GENERALES

L'entité PROGRAMME contient :

- . une définition, obligatoire, donnant ses caractéristiques générales (nom en clair, mots-clés, variante de COBOL à générer, etc.).
- . des lignes de Documentation généralisée, permettant notamment de donner des renseignements techniques (nom du programmeur, etc.).
- . des lignes de description, de plusieurs types :
 - > appels de Structures de Données assurant la génération de la 'DATA DIVISION' du Programme,
 - > langage structuré assurant la génération de la 'PROCEDURE DIVISION' du Programme,
 - > appels de Macro-Structures, paramétrées ou non, permettant de standardiser certains traitements effectués plusieurs fois dans les Programmes.

Les PROGRAMMES ainsi décrits permettront :

- . la génération de source COBOL adapté au système d'exploitation du matériel utilisé et prêt à être compilé,
- . une portabilité améliorée, l'adaptation au système d'exploitation s'effectuant par la modification d'une seule zone,
- . une adéquation entre les données décrites et le COBOL généré, puisqu'ils sont issus de la même source : la base de spécifications,
- . avec le module LANGAGE BATCH, l'obtention de fonctions automatiques (contrôle, mise à jour, édition) dans les Programmes générés.

2.2. STRUCTURE D'UN PROGRAMME

STRUCTURE D'UN PROGRAMME

Tout Programme est organisé en un ensemble de traitements successifs, effectués ou non dans une itération (en batch) ou dans une exécution (en TP). Ce sont :

- . l'acquisition des données,
- . les contrôles,
- . les mises à jour,
- . les éditions,
- . la restitution des résultats.

Chacun de ces traitements se compose d'un ensemble de séquences homogènes d'instructions appelées "fonctions".

Deux principes complémentaires structurent le Programme :

- . Enchaînement linéaire de fonctions dans l'ordre logique de leur exécution, chacune exécutant une tâche fonctionnelle ou technologique du Programme. Chaque fonction est identifiée par un code de 0A à 99.
- . Structuration hiérarchique des traitements dans chaque fonction. La fonction peut être décomposée en sous-fonctions, chacune décomposée à son tour en sous-fonctions, et ainsi de suite.

Les fonctions et sous-fonctions se suivent dans l'ordre de leur code selon la séquence de classement EBCDIC (les lettres précédant les chiffres), quelle que soit la séquence de tri du matériel utilisé.

PRINCIPES DE GENERATION D'UN PROGRAMME COBOL

Les Programmes réalisés en PACBASE sont générés à la demande, dans la variante de COBOL correspondant au matériel et au compilateur auxquels ils sont destinés.

L'IDENTIFICATION DIVISION du COBOL est générée à partir de la ligne de définition du Programme. Elle peut être modifiée ('-B').

L'ENVIRONNEMENT DIVISION et la FILE SECTION sont générées à partir des appels de Structures de Données dans le Programme ('-CD'). Elles peuvent être complétées ou modifiées ('-B').

Les autres sections de la DATA DIVISION sont générées à partir d'appels de Structures de Données et peuvent être complétées ou modifiées ('-W').

La PROCEDURE DIVISION est générée à partir des appels de Structures de Données ou Segments et des descriptions des traitements en Langage Structuré (-P).

Les lignes d'appel de Macro-Structures permettent d'appeler toutes les autres lignes de Langage Structuré pré-décrites (voir le chapitre "Macro-Structures").

CONSTANTES PACBASE

Le Système génère, dans tous les Programmes, en WORKING-STORAGE SECTION, une zone CONSTANTES-PAC dans laquelle sont déclarés en VALUE :

- le code de la bibliothèque où est défini le programme,
- le numéro de session de génération du programme,
- la date de génération du programme,
- le code du programme,
- le code de l'utilisateur ayant demandé la génération,
- l'heure de génération du programme,
- le nom externe du programme,
- le code de la base,
- la date de génération du programme avec le siècle.

Ces zones peuvent être employées dans le compte-rendu d'exécution du programme. Elles sont précédées du littéral "WORKING", qui peut servir de repère dans un "dump" en cas de problème à l'exécution.

2.3. DEFINITION D'UN PROGRAMME

DEFINITION D'UN PROGRAMME

L'écran de définition du Programme permet de lui attribuer un code en Bibliothèque et un nom en clair et de préciser ses principales caractéristiques.

Il s'obtient par le choix :

CH: P.....

REMARQUE :

Dans les descriptions d'écrans ou de bordereaux qui suivent, les codifications d'appel sont données pour le TP. Les codifications batch sont indiquées dans le chapitre "Répertoire d'accès", sous-chapitre "Batch".

```
-----  
!  
!          SUIVI PERMIS DE CONSTRUIRE          P0*DOC.LURE.JIA.893 !  
!  
! FICHE DU PROGRAMME      1  JIPED1          !  
!  
! NOM DU PROGRAMME.....: 2 EDITION PAR ANNEE COMM. ARRETE !  
!  
! CODE CLASSEMENT DU PROGRAMME.....: 3 JIPED1          !  
!  
! VARIANTE DU LANGAGE A GENERER.....: 4 0          !  
! OPTION NUMEROTATION CADRAGE COBOL..: 5          !  
! OPTION CARTES AVANT PROGRAMME.....: 6          !  
! OPTION CARTES APRES PROGRAMME.....: 7          !  
! CODE DU PROGRAMME GENERE.....: 8 JIPED1SP          !  
! TYPE DE PROGRAMMATION.....: 9 P          !  
! NATURE DU PROGRAMME.....:10 B          !  
! TYPE DE L'ENTITE.....:11 P  PROGRAMME          !  
! CONTROLE DE PRESENCE ZONE NUMERIQUE:12          !  
! GENERATION INDICATEURS SQL '_' : 13          !  
!  
!  
! MOTS CLES ASSOCIES....:14 STATISTIQUES DOSSIER          !  
!  
! NO DE SESSION.....: 0175          BIBLIOTHEQUE : JIA  BLOCAGE :          !  
!  
! O: C1 CH: Pjiped1          ACTION:          !  
!  
-----
```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| 1 | 6 | | CODE PROGRAMME (OBLIGATOIRE) Code identifiant le programme dans la bibliothèque. |
| 2 | 30 | | NOM EN CLAIR DU PROGRAMME (OBL. EN CREATION) Il doit être le plus explicite possible, car il permet la constitution automatique de mots-clés. |
| 3 | 6 | | CODE CLASSEMENT DU PROGRAMME GENERE (Valeur par défaut : code du Programme dans la Bibliothèque). Les Programmes sont classés dans l'ordre de ce code dans le flot généré. |
| 4 | 1 | | VARIANTE DU LANGAGE A GENERER Précise à quelle variante de langage il faut adapter les Programmes générés. Par défaut à la création, cette zone est initialisée à la valeur déclarée sur la Définition de la Bibliothèque dans la zone "LANGAGE GENERE". |
| | | N | Pas d'adaptation à une variante de langage Permet d'empêcher toute génération. |
| | | 0 | Adaptation au COBOL ANS IBM MVS |
| | | 1 | Adaptation au COBOL ANS IBM VSE |
| | | 2 | Adaptation au COBOL ANS IBM 36 |
| | | 3 | Adaptation au COBOL MICROFOCUS, IBM VISUAL SET |
| | | 4 | Adaptation au COBOL DPS7 |
| | | 5 | Adaptation au COBOL ANS 74 DPS8 |
| | | 6 | Adaptation au COBOL BCD DPS8 |
| | | 7 | Adaptation au COBOL HP-3000 |
| | | 8 | Adaptation au COBOL UNISYS Série A |
| | | 9 | Adaptation au COBOL ANS UNISYS 90/30 |
| | | A | Adaptation au COBOL 74 PRIME |
| | | B | Adaptation au COBOL UNISYS Série V |
| | | C | Extraction de source COBOL (voir le Manuel de Référence LANGAGE STRUCTURE, Chapitre "Annexes", Sous-cha- |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE (titre "Source COBOL") |
|-----|-----|------------------|--|
| | | D | Adaptation au COBOL ANS 74 CONTROL DATA |
| | | E | Adaptation au COBOL ANS 68 CONTROL DATA |
| | | F | Adaptation au COBOL TANDEM |
| | | I | Adaptation au COBOL DEC/VAX VMS |
| | | J | Adaptation au COBOL ANS PERKIN-ELMER-7-32 |
| | | K | Adaptation au COBOL ANS ICL 2900 |
| | | M | Adaptation au COBOL DPS6 |
| | | O | Adaptation au COBOL AS 400 |
| | | R | Adaptation au COBOL IBM 34 |
| | | S | Adaptation au COBOL SFENA |
| | | T | Adaptation au COBOL ANS SIEMENS |
| | | U | Adaptation au COBOL ANS 74 UNISYS série 1100 |
| | | V | Adaptation au COBOL ANS UNISYS 90/60 |
| | | W | Adaptation au COBOL DPPX IBM 8100 |
| | | X | Adaptation au COBOL ANS IBM MVS VS Cobol II |
| | | Y | Adaptation au COBOL IBM 38, AS 400 |
| 5 | 1 | | OPTION NUMEROTATION ET CADRAGE COBOL |
| | | | Permet de supprimer la numérotation, l'identification du Programme ou de modifier le cadrage des lignes du Programme généré. |
| | | BLANC | Numérotation, cadrage et identification du Programme conformes à la ligne COBOL standard. (Valeur par défaut). |
| | | 1 | Suppression de la numérotation. |
| | | 2 | Suppression de la numérotation et cadrage des instruc- tions (colonnes 8 à 71 incluses) en colonne 1. |
| | | 3 | Numérotation et cadrage standard, suppression de l'identification du Programme. |
| | | 4 | Suppression de la numérotation et de l'identification du Programme. |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | 8 | <p>-----</p> <p>Programmation par ligne de type '-8': réservé exclusivement à la maintenance d'applications écrites à l'aide de versions antérieures à la version 6 de PACBASE.</p> <p>La valeur mentionnée sur la ligne de définition de la Bibliothèque est reportée par défaut sur la ligne de définition d'un Programme lors de sa création.</p> <p>Au niveau d'un Programme, le type de programmation ne peut être modifié.</p> <p>La combinaison de lignes '-P' et de lignes '-8' dans un même Programme, soit directement, soit par l'intermédiaire de Macro-Structures appelées, est refusée.</p> |
| 10 | 1 | | <p>NATURE DU PROGRAMME</p> <p>Elle définit la structure du Programme généré ou la nature du Programme en Bibliothèque.</p> <p>B</p> <p>Structure d'un Programme 'BATCH'. (Option par défaut).</p> <p>C'est la structure normale d'un Programme itératif :</p> <ul style="list-style-type: none"> . Début de l'itération (F05) . Fin de travail (F20) . Fin d'itération (F9099. GO TO F05). <p>S</p> <p>Pas de structure de Programme imposée :</p> <p>Module Langage structuré :</p> <p>Cette nature peut être employée pour décrire un 'SYSTEME GENERATION' en TDS, un 'SCHEMA' en IDS II,...</p> <ul style="list-style-type: none"> . Suppression des divisions COBOL. . Le Programme est exclusivement constitué à partir des lignes '-B', '-W' et des descriptions de fichiers de Structures de Données appelées par les lignes '-CD'. <p>Module Générateur COBOL :</p> <p>Cette nature est obligatoire pour un Programme de TYPE 'S'. Le Programme est constitué à partir de lignes '-W', '-P', '-SC', et '-CP'.</p> <p>T</p> <p>Structure d'un Programme transactionnel.</p> <p>Suppression de la notion d'itération c'est-à-dire :</p> <ul style="list-style-type: none"> . Pas de génération de début de l'itération (F05). |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | C | <p>. Pas de fin de travail (F20). . Pas de fin d'itération (F9099. GO TO F05).</p> <p>Structure d'un Programme transactionnel sous C.I.C.S macro-level.</p> <p>Suppression de la notion d'itération c'est-à-dire : . Pas de génération du début de l'itération (F05). . Pas de fin de travail (F20). . Pas de fin d'itération (F9099. GO TO F05.)</p> <p>En outre : .Génération en tête de procédure division de l'ins- truction : MOVE CSACDTA TO TCACBAR. .Génération en F9099 de : DFHPC TYPE=RETURN. .Pas de numérotation des lignes du programme généré.</p> |
| | | M | <p>Macro-structure paramétrée. (Valeur documentaire uniquement).</p> <p>Un Programme de ce type est destiné à être incorporé dans un autre programme et ne peut donc être généré seul.</p> |
| | | F | <p>Programme constitué de lignes '-9' et '-CD'. Cette nature permet de prendre en compte les lignes '-9' d'appel de description de structures de fichiers générées automatiquement selon les caractéristiques de la ligne d'appel de Segment associée. (Voir le sous-chapitre "Sources COBOL" dans le Manuel de Référence LANGAGE STRUCTURE).</p> |
| | | D | <p>Programme constitué de lignes '-CD', '-B', '-W' et '-9'. Cette nature permet d'obtenir la génération automatique de l'ensemble des divisions IDENTIFICATION ENVIRONNEMENT et DATA, la PROCEDURE DIVISION étant entièrement écrite par lignes '-9'.</p> |
| | | P | <p>Programme constitué de lignes '-CD', '-B', '-W' et '-P' ou '-8' pour lequel seules les divisions IDENTIFICATION, ENVIRONNEMENT et DATA seront générées automatiquement, la PROCEDURE DIVISION étant entièrement obtenue à l'aide des lignes '-P' ou '-8'.</p> |
| 11 | 1 | | <p>TYPE DE L'ENTITE PROGRAMME</p> <p>Cette zone est interprétée à l'écran et sur les états par un libellé. Elle est purement documentaire pour les Programmes; elle permet d'inhiber la ventilation des Macro-Structures.</p> |
| | | A | <p>Système TP</p> |
| | | M | <p>Macro-structure</p> |

| NUM | LON | CLASSE VALEUR N | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|-----------------------|--|
| | | N | Macro-structure non ventilée |
| | | G | Grille |
| | | P | Programme |
| | | S | Schéma |
| | | T | Transaction |
| | | V | Sous-schéma |
| | | U | Utilitaire |
| | | D | Sous-programme |
| 12 | 1 | | <p>CONTROLE DE PRESENCE ZONE NUMERIQUE</p> <p>Dans les Programmes de contrôle, la détermination de la présence des Rubriques numériques se fera suivant la valeur de ce code.</p> <p>BLANC Présence de la Rubrique si elle n'est pas à blanc. (Valeur par défaut).</p> <p>0 Présence de la Rubrique si elle n'est pas à zéro.</p> <p>L Présence de la Rubrique si elle n'est pas à low-value. Cette option concerne alors toutes les Rubriques.</p> |
| 13 | 1 | | <p>GENERATION INDICATEURS SQL AVEC '-'</p> <p>Possibilité d'obtenir les chaînages sur l'utilisation des indicateurs SQL dans le Langage Structuré.</p> <p>BLANC Génération des indicateurs SQL sous la forme : VXXNNCORUB.</p> <p>- Génération des indicateurs SQL sous la forme : V-XXNN-CORUB.</p> |
| 14 | 55 | | <p>MOTS CLES ASSOCIES</p> <p>Cette zone n'existe qu'en TP. Pour introduire des mots-clés en batch, utiliser le bordereau G.</p> <p>Elle permet la saisie de mots-clés explicites, le système créant automatiquement des mots-clés implicites à partir du nom en clair des entités.</p> <p>Les différents mots-clés doivent être séparés par au moins un blanc.</p> <p>La longueur maximum d'un mot-clé est de 13 caractères. Ceux-ci doivent être alphanumériques. Les caractères = et * sont interdits.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>Majuscules et minuscules sont équivalentes, mais attention aux minuscules accentuées (e et E sont équivalents, mais pas é et E).</p> <p>NOTE : Les caractères accentués et les caractères spéciaux peuvent toutefois être déclarés équivalents à une valeur interne pour faciliter la recherche des occurrences par mots-clés.</p> <p>Pour plus de détails, se référer au Manuel d'Exploitation-Partie II "Guide de l'Administrateur", chapitre "Gestion de la base", sous-chapitre "PARM : MAJ des Paramètres Utilisateur".</p> <p>Il est possible d'affecter dix mots-clés explicites au maximum par entité.</p> <p>Pour plus de détails, voir le Manuel de Référence DICTIONNAIRE, Chapitre "Mots-Clés" Sous-chapitre "Comment constituer le Thésaurus".</p> |

3. MACRO-STRUCTURES

3.1. MACRO-STRUCTURES : PRESENTATION

MACRO-STRUCTURES PARAMETREES

Les Macro-Structures Paramétrées (M.S.P.) permettent de standardiser des séquences de Langage Structuré afin de les utiliser :

- . plusieurs fois dans un même Programme,
- . dans plusieurs Programmes,

avec des variantes éventuelles.

DEFINITION

Une Macro-Structure Paramétrée se définit par un écran de définition de programme (P).

Elle peut contenir un ensemble quelconque de lignes de description '-B', '-W' et '-P', caractérisant une ou plusieurs séquences d'instructions communes à un nombre quelconque de Programmes.

Elle peut aussi être issue de REVERSE ENGINEERING et contenir des lignes '-W', '-SC' (issues de REVERSE) et '-P' (si on l'a modifiée). Alors elle n'est prise en compte que dans les programmes issus de REVERSE (nature 'S').

Contrairement à un sous-programme, une M.S.P. peut contenir des lignes d'instructions non consécutives. En revanche, il est tout à fait possible de faire appel à un sous-programme à l'intérieur d'une M.S.P.

PRINCIPES

L'appel d'une M.S.P. dans un Programme va provoquer, lors de l'édition et de la génération du Programme, l'insertion des lignes de la M.S.P. aux emplacements correspondant à leur indicatif, compte tenu de la résolution éventuelle des paramètres.

Ainsi, pour le programmeur et pour le compilateur, tout se passe comme si les instructions de la M.S.P. faisaient partie du Programme.

TYPES DE MACRO-STRUCTURES

Les M.S.P. sont destinées à décrire des traitements standard communs à plusieurs programmes ou répétés dans un même programme.

On peut les regrouper en grands ensembles :

- . Un ensemble de type "squelette" permettant de donner une structure standard aux programmes, en fonction des normes en vigueur. Des traitements technologiques liés au moniteur de temps réel ou au SGBD utilisé peuvent y être incorporés.
- . Un ensemble de type "fonctions technologiques" permettant de standardiser des commandes spécifiques telles que les entrées-sorties d'un système de gestion de base de données (lecture, modification, suppression, création, etc.).
- . Un ensemble de type "fonctions techniques élaborées" dont le rôle est de résoudre a priori des traitements complexes (parcours d'une base de données, gestion des erreurs d'une application, lecture dynamique d'un fichier...).

C'est en général une combinaison de fonctions techniques élémentaires qu'il faut redécrire de façon à minimiser le travail nécessaire à la liaison de M.S.P. élémentaires entre elles.

- . Un ensemble de type "fonctionnel général". Ces M.S.P. décrivent des traitements communs à un ensemble d'applications tels que contrôle de date, calcul de lettre clé, transformation de date en délai... Elles sont indépendantes du matériel ou des outils utilisés.
- . Un ensemble de type "fonctionnel spécifique" permettant de définir des traitements communs à tous les programmes d'une application (par exemple, harmonisation de présentation des états ou des écrans de saisie).
- . Un ensemble permettant de réaliser des chaînages; par exemple si l'appel d'un sous-programme est fait par une M.S.P., on peut connaître automatiquement les programmes qui l'utilisent.

DIFFERENCE AVEC LES SOUS-PROGRAMMES

L'utilisateur est souvent amené à décider si des traitements communs à plusieurs Programmes doivent être regroupés dans un sous-programme ou dans une M.S.P. Pour résoudre ce problème, plusieurs questions doivent être posées :

- . les traitements communs sont-ils consécutifs ?
- . l'emplacement de ces traitements est-il défini ?
- . le nombre de paramètres est-il important ?
- . les traitements à effectuer sont-ils exécutés dans le cas général ?

Une fois toutes ces questions résolues, il sera possible de choisir quels traitements sont à effectuer dans des sous-programmes ou dans des M.S.P. :

- . s'ils ne sont pas consécutifs ---> M.S.P.
- . si l'emplacement est défini a priori ---> M.S.P.
- . si le nombre de paramètres est important ---> S.P.
- . si les traitements ne sont pas généraux ---> S.P.

PARAMETRAGE DE L'INDICATIF

Le Système permet de paramétrer la majeure partie de l'indicatif des lignes '-P' (fonction, sous-fonction et deux premiers caractères du numéro de ligne). Il est recommandé, avant d'écrire une macro-structure, d'essayer de structurer les traitements à écrire, c'est-à-dire de minimiser le nombre de paramètres de l'indicatif, de façon à :

- . faciliter l'utilisation de la M.S.P.,
- . éviter la prolifération de la même M.S.P. dans un seul Programme,
- . obtenir une structuration homogène des Programmes résultants.

NOMBRE DE PARAMETRES

Dans tous les cas, il est recommandé de minimiser le nombre de paramètres d'une M.S.P., de façon à :

- . faciliter l'utilisation de la M.S.P.
- . minimiser le temps de résolution de la M.S.P.

Il est souvent préférable de 'banaliser' certains codes (fichier, Segment, Rubrique) dans une zone de travail spécifique de la M.S.P., définie par l'appel de Rubriques en zones de travail.

DOCUMENTATION D'UNE M.S.P.

Les M.S.P. peuvent être documentées à l'aide des lignes de Documentation Généralisée (-G) et des Textes ventilés. On peut également documenter une M.S.P. par des lignes d'appel de Macro-Structures avec son propre code comme code Programme appelant et comme code Programme appelé. En particulier, il est souhaitable de définir de cette façon les paramètres utilisés car ces lignes figurent dans la description des Programmes appelant la Macro-Structure.

MODIFICATION D'UNE LIGNE DE M.S.P.

A indicatif égal les lignes du Programme appelant sont prioritaires sur les lignes de la M.S.P. Il est souhaitable de concevoir les M.S.P. de façon à ce qu'il n'y ait pas de lignes modifiées par Programme.

La maintenance des deux entités en sera facilitée. Si des insertions de lignes spécifiques Programme sont nécessaires, elles doivent être soigneusement documentées.

COHERENCE DES PARAMETRES

Un Programme fait souvent appel à plusieurs M.S.P. Dans ce cas, il est souhaitable que la codification des paramètres soit cohérente, c'est-à-dire que des informations de même type apparaissent dans le même ordre. Par exemple, si deux M.S.P. appelées dans un Programme ont comme paramètre un code fichier, il est souhaitable qu'il ait le même rang dans les deux M.S.P. Cela n'est pas toujours possible, mais en écrivant une M.S.P., on essaiera d'étudier les M.S.P. déjà existantes afin de minimiser les différences.

REMARQUE :

Tout Programme peut être utilisé comme Macro-Structure non paramétrée, à condition que son code soit classé avant le code du programme appelant (dans l'ordre EBCDIC).

3.2. APPEL DE MACRO-STRUCTURES (-CP)

APPEL DE MACRO-STRUCTURES (-CP)

L'appel de Macro-Structures permet d'insérer dans un Programme les lignes définies dans une Macro-Structure en précisant les valeurs par lesquelles les paramètres éventuels doivent être remplacés.

PARAMETRAGE

Il est possible de déclarer 20 paramètres pour chaque M.S.P. Les 10 premiers paramètres sont indiqués par \$n (n = 1,2,... 9,0) sur la première ligne ; les 10 suivants (n = A,B,...J), doivent figurer sur une ligne "suite", portant le même numéro que la précédente. Les valeurs alphabétiques ne doivent pas être utilisées pour paramétrer les indicatifs des lignes de la Macro-Structure.

Les indicatifs de lignes sont: des codes Fonctions et Sous-Fonctions, ainsi que des numéros de lignes.

Chaque occurrence d'un paramètre \$n dans la Macro-Structure sera remplacée dans le Programme appelant par une chaîne de caractères quelconque.

Le nombre de caractères de chaque valeur de remplacement est limité :

- . à cause de la longueur des Rubriques des lignes de programmation dans lesquelles figure le paramètre remplacé.
- . par le fait que toutes les valeurs de remplacement doivent tenir, avec les délimiteurs, sur deux lignes d'appel.

OCCURRENCES D'ENTITES UTILISEES COMME PARAMETRES

Vous pouvez utiliser des occurrences de Rubrique, Structure de Données et Segment comme paramètres.

Lorsqu'on appelle une de ces occurrences par l'intermédiaire des paramètres, aucun chaînage n'est établi si le nom de l'occurrence est déclaré comme une simple chaîne de caractères.

Il est possible d'obtenir ce chaînage en spécifiant, dans les valeurs attribuées aux paramètres, que la valeur est un code Rubrique, Structure de Données ou Segment en codant, respectivement :

/E=Corub/ ou /D=dd/ ou /S=segt/

Lors de la transformation, le paramètre est remplacé par Corub, dd ou segt et les chaînages sont établis.

MACRO-STRUCTURES NON-VENTILEES

Un 'N' dans la colonne Ventilation ('V') indique les Macro-Structures non ventilées.

SAISIE DES COMMENTAIRES

Les lignes de commentaires saisies sur l'écran -CP sont visualisées uniquement à partir du Programme appelant. Il est possible d'indiquer à la suite du code de la macro-structure le numéro de la ligne à partir de laquelle doit commencer l'affichage.

En revanche, les lignes de commentaires saisies sur l'écran -XP de la M.S.P. elles seront visualisées lors de l'appel de la M.S.P sur cet écran.

La saisie de commentaires sur l'écran -XP permet ainsi de faciliter la saisie des paramètres lors des appels de la M.S.P.

```
-----  
!                                     !  
!           SUIVI PERMIS DE CONSTRUIRE           P0*DOC.LURE.JIA.893 !  
!                                     !  
! MACRO-STRUCTURES DU PROGRAMME           1 JIPED1 EDITION PAR ANNEE COMM. ARRETE !  
!                                     !  
!  2   3   4   5   6                                     7   !  
! A  MACRO NL S : SIGNIFICATION OU VALEUR DES PARAMETRES           D V !  
!   AAED12 :                                           !  
!   AAPR20 :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
!           :                                           !  
! O: C1 CH: P jiepd1 CP                                           !  
!                                     !  
-----
```

PROGRAMMES/ECRANS APPELANT LE PROGRAMME (-XO/-XP)

De même que l'écran -CP, cet écran permet d'appeler des Macro-structures, et d'indiquer leurs paramètres,

Il permet de plus de visualiser les commentaires affectés à la Macro-structure.

Il dispose donc d'un type de ligne indiquant s'il s'agit :

- . d'un commentaire (type 'C'),
- . d'un appel de programme (type 'P') ou d'écran (type 'O').

Il est possible d'indiquer le numéro de ligne de l'appel de la macro-structure immédiatement après le nom du Programme ou de l'Ecran à partir duquel doit commencer l'affichage.

```
-----  
!                                     !  
!          SUIVI PERMIS DE CONSTRUIRE          P0*DOC.LURE.JIA.893 !  
! PROGRAMMES/ECRANS APPELANT LE PROGRAMME 1 AVJIA1 AVIS, CALCUL DATE LIMITE !  
!                                     !  
! 2           4 5 6                               7           !  
! A T PG/EC  NL S : SIGNIFICATION OU VALEUR DES PARAMETRES          D V !  
! C           10 : $1=VALEUR MINIMALE DU CODE AVIS                    !  
! C           20 : $2=RUBRIQUE A CONTROLER                            !  
! O JIE020    : 020/E=CODAV/                                          !  
! O JIE050    : 050/WWORG/                                            !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
!           : :                                                         !  
! O: C1 CH: P avjialXO                                               !  
!                                     !  
-----
```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| 1 | 6 | | CODE PROGRAMME OU ECRAN (OBLIGATOIRE) |
| 2 | 1 | | CODE ACTION |
| 3 | 6 | | <p>CODE DE LA MACRO-STRUCTURE (OBLIGATOIRE)</p> <p>Une Macro-Structure est un ensemble quelconque de lignes '-B', '-W', '-P' ou '-W', '-P', et '-SC' (si elle est issue de REVERSE ENGINEERING).</p> <p>Ce n'est pas un sous-programme, mais une séquence de langage structuré incorporée dans les Programmes avant génération. Lors de cette incorporation, si une ligne de la Macro-Structure a le même indicatif qu'une ligne du programme, elle ne sera pas prise en compte et sera repérée dans le programme appelant par le caractère '*' dans la zone 'CODE ACTION'.</p> <p>Si des lignes de deux Macro-Structures ont le même indicatif, elles sont toutes ignorées à la génération du programme appelant.</p> <p>Les lignes de la Macro-Structure remplacées par des lignes du programme ayant le même identifiant sont considérées comme des commentaires; elles apparaissent (dans la limite de 10) dans le descriptif du Programme appelant.</p> <p>Une Macro-Structure non paramétrée peut comporter des lignes d'appel de Structure de Données (non incorporée dans les Programmes), et peut être générée seule.</p> <p>Le code de la Macro-Structure ne peut être supérieur à celui des Programmes qui l'appellent (classement EBCDIC).</p> <p>La ligne d'appel est rattachée à la Macro-Structure : la suppression de la Macro-Structure entraîne celle des lignes d'appel, la suppression d'un Programme n'entraîne pas celle des lignes d'appel qui le concernent.</p> |
| 4 | 2 | | <p>NUMERO DE LIGNE</p> <p>ZONE NUMERIQUE PURE (Blancs et zéros équivalents).</p> <p>Il permet de définir plusieurs lignes de commentaire pour une Macro-Structure, ou lorsque la Macro-Structure est paramétrée, de l'appeler plusieurs fois dans le même Programme, avec des valeurs différentes pour les paramètres.</p> |
| 5 | 1 | | SUITE |
| | | * | Création d'une ligne suite. |
| 6 | 50 | | COMMENTAIRES, VALEURS DES |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | <p>PARAMETRES</p> <p>Appel de Macro-Structures (-CP) : Valeurs des paramètres dans le cas d'une Macro-Structure Paramétrée si cette zone contient des '/', commentaire dans tous les autres cas.</p> <p>Programmes/Ecrans appelant le Programme (-XP/XO) : Valeurs des paramètres sur une ligne de type P ou O. Commentaires sur une ligne de type C.</p> <p>Les valeurs des paramètres doivent être indiquées les unes à la suite des autres, terminées par un '/', dans l'ordre des numéros de paramètre, le premier '/' indiquant la fin de la valeur du paramètre \$1, le dixième la fin de celle de \$0.</p> <p>Deux '/' consécutifs permettent d'ignorer le paramètre correspondant.</p> <p>Il est possible de créer une ligne suite par appel si les valeurs ne tiennent pas sur la première ligne.</p> <p>Le '/' est pris par défaut comme délimiteur mais il peut être remplacé par un caractère quelconque indiqué dans la zone 'Délimiteur des valeurs de paramètres'.</p> <p>Lorsqu'une Rubrique, une Structure de Données ou un Segment est appelé comme paramètre, il est possible d'établir une référence croisée en codant, respectivement, l'appel: /E=CORUB/ (CORUB = code Rubrique), ou /D=DD/ (DD = code Structure de Données) ou /S=SEGT/ (SEGT = code Segment).</p> <p>Le paramètre est alors remplacé par la valeur CORUB, DD ou SEGT et les chaînage avec la Rubrique, la Structure de Données ou le Segment sont établis.</p> |
| 7 | 1 | | <p>DELIMITEUR DES VALEURS DE PARAMETRES</p> <p>C'est le caractère servant à séparer les différentes valeurs des paramètres. S'il n'est pas indiqué, le caractère '/' est pris par défaut.</p> |

3.3. MACRO-STRUCTURES NON VENTILEES

MACRO-STRUCTURES NON VENTILEES

De manière générale, les lignes d'une M.S.P. apparaissent dans la description du Programme appelant, comme si elles appartenait au programme.

Certaines M.S.P. peuvent être appelées un grand nombre de fois dans plusieurs Programmes. Le programmeur les considère alors comme faisant partie de son environnement.

Ces M.S.P. peuvent être définies comme non ventilées; les lignes ne sont alors pas visualisables à la consultation du programme.

Bien entendu, la M.S.P. est prise en compte lors de l'édition et la génération du Programme.

Les Macro-Structures non ventilées ne concernent que les Programmes batch.

AVANTAGES ET INCONVENIENTS

L'appel d'une M.S.P. dans un Programme provoque la création d'un index appelé "index de ventilation" pour chaque ligne de sa description.

Cet index est créé pour l'affichage des lignes de la M.S.P. dans la description du Programme appelant.

La ventilation d'une M.S.P. dans un Programme peut donc être pénalisante pour trois raisons :

- . Temps de réponse durant la mise à jour de la M.S.P. qui doit être ventilée, pour l'ensemble des utilisateurs ("sérialisation" des mises à jour),
- . Désorganisation du fichier index (AN) par l'insertion de masse de clés souvent contiguës,
- . Augmentation importante du nombre d'enregistrements du fichier index (AN), qui se répercute sur le temps d'exécution des procédures batch de sauvegarde, restauration et réorganisation.

Les M.S.P. non ventilées permettent d'améliorer ces trois points, dans la mesure où l'expansion de la M.S.P. n'étant pas faite en temps réel en mise à jour, elle ne crée pas d'enregistrement dans le fichier mais elle est résolue au moment de l'édition-génération du Programme appelant.

En contrepartie :

- . Une M.S.P. non ventilée n'étant pas affichée dans le Programme appelant en TP, l'écriture et la maintenance de ce Programme sont plus difficiles,
- . L'expansion étant faite lors de l'extraction du Programme pour édition-génération, le temps d'exécution de la procédure est augmenté.

MISE EN OEUVRE

Une M.S.P. non ventilée est une M.S.P. dont le type d'entité (saisi sur l'écran de définition de la M.S.P.), est 'N'.

Lors de l'appel de la M.S.P., la lettre 'N' s'affiche dans la zone VENTILATION ('V') de l'écran d'appel, indiquant ainsi les M.S.P. non-ventilées.

4. EMPLOI DES STRUCTURES DE DONNEES

4.1. APPEL DES STRUCTURES DE DONNEES (-CD)

APPEL DE STRUCTURES DE DONNEES DANS UN PROGRAMME (-CD)

L'appel des Structures de Données permet de générer tout ou partie de la DATA DIVISION et de la PROCEDURE DIVISION des programmes COBOL.

L'écran d'appel des Structures de Données est obtenu par le choix '-CD' pour le programme considéré (entité 'P').

CARACTERISTIQUES GENERALES

L'appel d'une Structure de Données peut comporter plusieurs lignes si nécessaire. Toutefois certaines informations ne peuvent figurer que sur la première ligne.

Par défaut, le Système affecte des valeurs aux informations obligatoires de la ligne. Dans ces conditions une Structure de Données sera considérée comme décrivant un fichier séquentiel avec des enregistrements de longueur fixe; le fichier comprendra tous les enregistrements de la Structure de Données, avec les Rubriques en format interne, sans les Rubriques facultatives.

ORGANISATION

Les appels de Structures de Données servent à décrire trois grands types de données, précisés dans la zone 'Organisation'. Ce sont :

- . des fichiers classiques,
- . des bases de données,
- . des zones de travail ou de lien.

Toutefois, on définira les zones de travail et de lien de préférence sur l'écran '-W' réservé à cet usage.

De plus, les lignes '-W' d'une macro-structure sont incorporées dans le programme appelant, mais pas les lignes '-CD'.

Il faut cependant noter que l'appel de Structures de Données en lignes -W ne permet pas la création de lignes suite (ce qui limite la sélection de Segments à quatre, par exemple), ni la déclaration de l'utilisation, de ruptures ou de synchronisation.

STRUCTURE DE DONNEES COMPOSITE

Il est possible, au niveau du Programme, de constituer une Structure de Données avec des Segments appartenant à des Structures de Données différentes.

Sur plusieurs lignes ayant le même code S.D. dans le Programme, on indique les codes des différentes Structures de Données avec sélection éventuelle de Segments.

La partie commune sera constituée par le code de la Structure de Données sélectionnée sur la première ligne.

Pour appeler des Segments qui ont le même code enregistrement (mais dont les codes Structure de Données bibliothèque sont différents) dans une Structure de Données Programme, il est nécessaire de changer le code enregistrement de l'un d'entre eux dans la zone argument.

```

-----
!
!           VALIDATION MATERIEL IBM                       PO*AG.LILI.IBM.847 !
!
! STRUCTURES DE DONNEES DU PROGRAMME 1 FO20MA MISE A JOUR FOURNISSEURS !
!
!  2  3  4    5  6      7- 11 12    13 14 16    18 19    21      23 25 27 !
!
!           15 17      20    22      24 26 !
! A PR SU : SD EXTERN OAMOU BLOC T R S U RE SE M UNIT P SELECTION F O D N E !
! ED      : ED ED      SSFOU  0 R      I      1      I      1      !
!           : Z.COMPL.: 28      CLE ACC.: 29      IDENT. : 30      !
! AC      : AC FORACH VSFID  0      T      CLE ACC.: NUACH      IDENT. :      !
!           : Z.COMPL.: AC00STATU CLE ACC.:      IDENT. :      !
! CA      : AC SEQACH SSFOU  0      D      IDENT. :      I      1      !
!           : Z.COMPL.:      CLE ACC.:      IDENT. :      !
! FB      : FB FORFOU VRFRD  0      1 P FB      F      I      1      !
!           : Z.COMPL.: FB00STATU CLE ACC.: NUFOU      IDENT. :      !
! MF      : MF MVTFOU SSFIU  0      M      FDAS      E      1      !
!           : Z.COMPL.:      CLE ACC.:      IDENT. :      !
! TA      : TA XTAB  GRFIU  0      C      IDENT. :      I      2 1      !
!           : Z.COMPL.:      CLE ACC.:      IDENT. :      !
! XE      : XE EDICR  SSFOU  0      I      S      I      1      !
!           : Z.COMPL.:      CLE ACC.:      IDENT. :      !
! XS      : XS FORXS  VRFID  0      C      IDENT. :      I      1      !
!           : Z.COMPL.: XS00STATU CLE ACC.: XCLER7      IDENT. :      !
!           :      CLE ACC.:      IDENT. :      !
!           : Z.COMPL.:      CLE ACC.:      IDENT. :      !
!
! O: C1 CH: -CD
!
-----

```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| 1 | 6 | | CODE PROGRAMME (OBLIGATOIRE) Code identifiant le programme dans la bibliothèque. |
| 2 | 1 | | CODE ACTION (OBLIGATOIRE) |
| 3 | 2 | ALPHA. | CODE STRUCTURE DONNEES PROGRAMME (OBLIGATOIRE) Ce code établit la séquence de traitement de la S.D. dans le Programme. Il doit être alphabétique. Il est conseillé, lorsqu'une S.D. décrite en Bibliothèque n'est utilisée qu'une fois dans un Programme, de prendre comme code dans le Programme le code en Bibliothèque. |
| 4 | 2 | ALPHA. | SUITE Coder deux lettres pour indiquer une ligne suite. Cette zone doit être laissée à blanc pour la première ligne d'appel de la S.D., qui définit les caractéristiques techniques, cinématiques et descriptives de la S.D. Les lignes SUITE permettent de sélectionner des enregistrements de diverses S.D. de la Bibliothèque et éventuellement de demander leur description à un emplacement précis. |
| 5 | 2 | | CODE STRUCTURE DE DONNEES Code alphanumérique. Il s'agit d'un code logique propre à la Base et donc indépendant des noms utilisés dans les Blocs Bases de Données et dans les Programmes. |
| 6 | 6 | | NOM EXTERNE DE LA S.D. Valeur par défaut : code S.D. programme. POUR L'ORGANISATION Y : Cette zone doit contenir le code VisualAge Pacbase du Serveur qui accède à la Vue Logique. Pour plus d'informations, référez-vous au Manuel de Référence PACBENCH CLIENT/SERVEUR. POUR LES ORGANISATIONS SQL : Cette zone doit contenir le code VisualAge Pacbase du bloc SQL. Pour plus d'informations, référez-vous au Manuel "Langage Structuré", Chapitre "Description des Traitements (-P)" et au Manuel "Description de bases de |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | <p>données relationnelles", Chapitre "Accès SQL", Sous-chapitre "Personnalisation des accès SQL".</p> <p>POUR LES AUTRES ORGANISATIONS :</p> <p>IBM OS (variante 0) : DDNAME sur 1 à 6 positions.</p> <p>IBM DOS (variante 1) trois formes :</p> <ul style="list-style-type: none">. SYSnnn nom symbolique d'unité,. XXXnnn précise à la fois l'unité symbolique SYSnnn et le nom externe du fichier XXX,. XXXXXX nom externe, l'unité symbolique est générée avec SYSnnn, nnn étant incrémenté de 1 pour chaque fichier en partant de SYS010. <p>BULL DPS7 (variante 4) : INTERNAL-FILE-NAME sur 1 à 6 positions.</p> <p>BULL DPS8 ASCII (variante 5) : File-code (2 car.).</p> <p>BURROUGHS large system (variante 8), UNISYS série A (variante 8) : NNPPP numérique, fait générer AREA nn AREASIZE pppp.</p> <p>CDC (variantes D et E) : indiquer OUTPUT pour une imprimante, nom externe sur 1 à 6 positions dans les autres cas.</p> <p>BULL DPS8 BCD (variante 6) : 2 lettres.</p> <p>PRIME (variante A) : pris en compte si le type d'unité est D ou U.</p> <p>BURROUGHS médium system (variante B), UNISYS série V (variante B) : ne correspond pas au nom externe, mais à l'espace occupé sur disque:</p> <ul style="list-style-type: none">. nombre de zones sur 2 caractères numériques,. nombre d'enregistrements par zone sur 4 caractères numériques. <p>PERKIN-ELMER (variante J) : avec l'organisation S indiquer LUnn, nn variant de 00 à 99.</p> <p>CII-HB MINI-6 (variante M) : 2 chiffres ou lettres.</p> <p>SIEMENS (variante T) : SYS000 à SYS099.</p> <p>IBM VS2 (variante X) : DDNAME sur 1 à 6 positions.</p> <p>TANDEM (variante F) : nom externe sur 1 à 6 positions.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE DEC/VMS (variante I): nom externe sur 1 à 6 positions. |
|-----|-----|------------------|---|
| | | | CARACTERISTIQUES PHYSIQUES FICHIER |
| 7 | 1 | | ORGANISATION |
| | | S | Séquentielle (Valeur par défaut). |
| | | I | Séquentielle indexée (ISP pour DPS8 BCD). Pour un fichier ISP de code LE, seront générées en WORKING 3 zones : LE-FICHIER LE-DATA et INVKEY. LE-DATA aura pour VALUE le nom externe qui devra être le FILE CODE de la ligne \$ DATA précédant, dans le flot des lignes contrôle d'exécution, les lignes ISP donnant les caractéristiques physiques du fichier. |
| | | V | VSAM (IBM), UFAS (CII-HB), etc. Fait générer la clause FILE STATUS et la zone correspondante, à condition que celle-ci soit déclarée dans la zone complémentaire. Le fichier est déclaré séquentiel si le nom de la clé dans l'enregistrement est absent, indexé s'il est renseigné. |
| | | W | Les descriptions sont générées en WORKING, avant la constante 'DEBUT-WSS'. Une S.D. ainsi décrite sera utilisée comme zone de travail ou traitée à travers une fonction d'un système de gestion généralisé (bases de données par exemple). |
| | | L | Identique à 'W', l'emplacement de la description étant laissé au choix de l'utilisateur (Voir la zone EEMPLACEMENT DE LA DESCRIPTION). |
| | | X | S.D. mentionnée à titre de commentaire, ne donne lieu à aucune génération. |
| | | G | Description de table. Provoque la génération de la zone de communication avec le module d'accès généralisé (Cf. Manuel de Référence TABLES). |
| | | Y | Appel de la CLAUSE COPY qui correspond à la zone de communication entre le client et le serveur. Pour plus d'informations, référez-vous au Manuel de Référence PACBENCH CLIENT/SERVEUR. |
| | | | BASES DE DONNEES ----- Les valeurs de code suivantes sont réservées à la description des bases de données pour les utilisateurs ne |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | disposant pas du module 'Description de Bases de Données' et à leur utilisation dans les programmes d'application dans tous les cas. |
| | | A | Réservé à la description des fichiers ADABAS dans les programmes de définition ou d'utilisation de la base de données. |
| | | B | Réservé à la description des records d'une base de données IDMS dans les sous-schémas ou les programmes d'application. |
| | | D | Réservé à la description des segments ou records de différentes bases de données, DL/1, IDS I, IDS II, (suivant la variante du programme), dans la génération de DBD, SYSGEN, schémas ou programmes d'application (suivant la nature du programme). |
| | | F | Réservé à la description des FORMAL d'une base de données Socrate/Clio. |
| | | Q | (IBM) description de bases de données SQL/DS, DB2/2 ou DB2/6000. (HP 3000) description de bases de données ALLBASE/SQL (MICROFOCUS) description de bases de données DB2/2 ou DB2/6000. |
| | | T | Reservé à la description des fichiers 'TOTAL' dans les programmes de définition ou d'utilisation de la base de données. |
| | | Z | Réservé à la description des structures d'une base de données Socrate/clio. |
| | | 2 | Génération-Description d'un segment DB2 ou VAX/SQL. Seuls les accès physiques ne sont pas générés. La structure des indicateurs variables correspondant aux colonnes de la table DB2 ou VAX/SQL est toujours générée. |
| | | C | Réservé à la description des structures d'une base de données INTEREL RDBC ou RFM. |
| | | O | Réservé à la description des structures d'une base de données ORACLE (<V.6) |
| | | P | Réservé à la description des structures d'une base de données ORACLE (V.6 et V.7) |
| | | R | Réservé à la description des structures d'une base de données RDMS. |
| | | 3 | Réservé à la description des structures d'une base de |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE données SQL SERVER. |
|-----|-----|------------------|--|
| | | 4 | Réservé à la description des structures d'une base de données DB2/400. |
| | | N | Réservé à la description des structures d'une base de données NONSTOP SQL. |
| | | M | Réservé à la description des structures d'une base de données DATACOM DB. |
| | | 9 | Réservé à la description des structures d'une base de données INFORMIX, INGRES/SQL, ou SYBASE. |
| | | | L'utilisation du Système avec les SGBD est développée dans des manuels spécifiques, auxquels l'utilisateur se reportera pour de plus amples informations. |
| 8 | 1 | | MODE D'ACCES |
| | | S | Séquentiel (option par défaut). |
| | | R | Direct. |
| | | D | Dynamique (Reservé à l'organisation V). |
| 9 | 1 | | MODE DE BLOCAGE |
| | | C | Pour une organisation de type P (Oracle V6 et V7) ou 9 (Sybase) : génération automatique des CONNECT AT database, DECLARE database et accès SQL AT database. |
| | | F | Fixe (option par défaut). Lors de la génération, les longueurs des différents enregistrements seront alignées sur la longueur du plus grand. |
| | | V | Variable. |
| | | U | Indéfini. |
| | | S | SPANNED (réservé aux variantes IBM OS et DOS). |
| 10 | 1 | | OUVERTURE |
| | | I | Entrée, option par défaut pour fichiers d'utilisation: C, T, X, M, N, P, interdite pour les autres. |
| | | O | Sortie, option par défaut pour fichiers d'utilisation: D, S, R, E, I, J, interdite pour les autres. |
| | | E | Sortie. Génération de la clause OPEN EXTEND (uniquement pour les variantes de langage : 2, 4, 5, 6, D, E, F, G, H, I, J, K, Q, S, U, W, X, Y). |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | T | A trier (en entrée ou sortie suivant utilisation). |
| | | R | Entrée-sortie (fichiers en accès direct seulement). |
| 11 | 1 | | TYPE D'UNITE |
| | | U | Mémoire magnétique à accès séquentiel. (Valeur par défaut). |
| | | D | Mémoire magnétique à accès sélectif. |
| | | R | Périphériques lents (lecteur perforateur imprimante). |
| | | | La codification de ce paramètre est importante dans les variantes de COBOL où les clause ASSIGN, les niveaux FD ou les ordres WRITE dépendent du type d'unité choisi. |
| 12 | 5 | NUMER. | FACTEUR DE BLOCAGE BLANCS ET ZEROS EQUIVALENTS |
| | | | ZONE NUMERIQUE PURE |
| | | | Option par défaut : 0. |
| | | | Le blocage peut être nul dans les cas suivants : |
| | | | . IBM OS (variante 0), sauf pour les fichiers en organisation indexée, |
| | | | La clause COBOL correspondante (BLOCK CONTAINS) n'est pas générée dans les cas suivants : |
| | | | . Fichier tri. |
| | | | . Fichier disque si nombre non mentionné. |
| | | | . Fichier sur type d'unité R en IBM DOS (variante 1) et PRIME (variante A). |
| | | | . Fichier fixe ou avec block 0 pour BULL DPS8 BCD (variante 6) |
| | | | . Block 0 pour CDC (variantes D et E), Burroughs Large System/Unisys série A (variante 8), IBM 8100 (variante W), IBM 36 (variante 2), IBM 38 (variante Y), AS 400 (variante O). |
| | | | Rubrique inutilisée pour ICL 2900 (variante K) et Perkin-Elmer (variante J). |
| 13 | 1 | | TYPE DE BLOCAGE |
| | | R | Records (valeur par défaut). |

| NUM | LON | CLASSE VALEUR C | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE Characters. |
|-----|-----|-----------------------|---|
| 14 | 1 | 0 à 9 | <p>NOMBRE DE NIVEAUX DE RUPTURE</p> <p>(Module LANGAGE BATCH)</p> <p>L'argument (ou indicatif) d'une Structure de Données se compose, au niveau d'un Programme, de 9 éléments au maximum (groupes ou Rubriques élémentaires). Il est défini dans la ligne d'appel de Structures de Données (voir argument).</p> <p>Ces 9 éléments définissent 9 niveaux de rupture de séquence au maximum. Le niveau indiqué est le niveau mineur sur lequel on désire déceler ces ruptures. (Option par défaut 0).</p> <p>Les ruptures sur les niveaux majeurs seront automatiquement décelées. Le tout sera disponible sous forme de deux jeux de variables :</p> <p>. dd-PEn : premier Segment sur la S.D. dd au niveau n (valeur 1 oui, 0 non),</p> <p>. dd-DEn : dernier Segment sur la S.D. dd au niveau n (valeur 1 oui, 0 non),</p> <p>Lorsque la S.D. est synchronisée avec d'autres, les ruptures seront automatiquement calculées sur l'ensemble des S.D. et disponibles sous forme de deux jeux de variables indiquant le changement de l'indicatif de niveau n (ou rupture de niveau n) :</p> <p>. RTPn pour la rupture entre l'itération précédente du Programme et celle en cours,</p> <p>. RTDn pour la rupture entre l'itération en cours du Programme et la prochaine,</p> <p>Ces variables sont également disponibles lorsqu'il n'y a pas de S.D. synchronisées dans le Programme.</p> <p>Pour une S.D. d'utilisation 'D', affecter un nombre de niveaux de rupture différent de zéro permet d'obtenir au niveau du Programme généré une zone de réserve en WORKING affectée du radical '1-', qui sera utilisée pour les traitements sur le fichier.</p> |
| 15 | 1 | 0 A 9 | <p>NOMBRE DE NIVEAUX DE SYNCHRONISATION</p> <p>Les S.D. en accès séquentiel en entrée lues simultanément doivent être synchronisées. Le niveau indiqué est le niveau mineur de l'argument sur lequel les S.D. sont rapprochées. Lorsque la synchronisation se limite à deux S.D., les niveaux doivent être égaux (non con-</p> |

| NUM | LON | CLASSE VALEUR | <p>SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE</p> <p>trôlé). Lorsqu'il y a plus de 2 S.D., certaines peuvent être synchronisées à un niveau majeur par rapport aux autres.</p> <p>(Valeur par défaut 0).</p> <p>Pour que la synchronisation soit automatique, les conditions suivantes doivent être réalisées :</p> <ul style="list-style-type: none"> . Le niveau de rupture des S.D. doit être égal au niveau de synchronisation - 1, sauf pour une S.D. Mouvement pour laquelle le niveau de rupture doit être supérieur ou égal au niveau de synchronisation. . Les Rubriques composant l'argument de chaque S.D. doivent être classées en séquence croissante. . Les Rubriques composant l'argument de chaque S.D. doivent être de même longueur à niveau égal. . Ces Rubriques doivent avoir un format étendu (si elles sont numériques elles doivent être entières et non signées). <p>Si ces conditions ne sont pas réalisées, la cinématique du Programme doit être modifiée par ligne -P.</p> <p>La synchronisation se traduit par deux jeux de variables donnant l'état des S.D. :</p> <ul style="list-style-type: none"> . dd-CFn configuration de la S.D. dd au niveau n indique la participation de la S.D. en entrée au traitement pour ce niveau d'indicatif. (Valeur 1 l'argument à ce niveau est égal au minimum en cours de traitement, valeur 0 sinon). . dd-OCn occurrence de la S.D. dd au niveau n, indique, pour les S.D. principales, que l'on est en cours de mise à jour d'un Segment ou d'une séquence de Segments de la S.D. dd. |
|-----|-----|----------------------------|--|
| 16 | 1 | <p>C</p> <p>D</p> <p>T</p> | <p>UTILISATION DE LA S.D.</p> <p>Code permettant de définir le rôle de la S.D. dans le Programme, et commandant la génération des fonctions automatiques du module Langage Batch.</p> <p>Consultée : S.D. quelconque en entrée.</p> <p>Divers: S.D. quelconque en sortie (valeur par défaut).</p> <p>Table : S.D. destinée à être stockée intégralement en mémoire. La table de stockage est générée en fonction du nombre de répétitions indiqué sur la ligne de définition du segment. Il n'est pas possible de sélection-</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | ner plus de 50 segments pour une S.D. d'utilisation 'T'. |
| | | X | Table = S.D. stockée partiellement en mémoire (seules les rubriques autres que FILLER sont décrites). Les rubriques élémentaires autres que FILLER, sont limitées à 10 pour la partie commune (code structure excepté) et à 29 pour chaque partie spécifique. |
| | | S | Sélectionnée: S.D. en sortie extraite d'une autre S.D. A la différence d'une S.D. 'D', sa description en zone de sortie n'est pas détaillée. Lorsque la S.D. contient des Rubriques dont le nombre de répétitions dépend d'un compteur (OCCURS DEPENDING ON), son utilisation doit être 'D'. Les valeurs suivantes sont spécifiques au Module LANGUAGE BATCH : |
| | | P | Principale : S.D. en entrée susceptible d'être mise à jour (par une S.D. d'utilisation M ou N). |
| | | R | Résultat : S.D. principale après mise à jour en accès séquentiel (si la S.D. contient des clauses OCCURS DEPENDING, le résultat doit être déclaré en utilisation 'D'). |
| | | M | Mouvements à contrôler : S.D. en entrée à contrôler pouvant mettre à jour d'autres S.D. (fonctions générées : 30 à 76). Il n'est pas possible de déclarer plus d'une S.D. d'utilisation M ou N par Programme. |
| | | N | Mouvements à ne pas contrôler: S.D. en entrée pouvant mettre à jour d'autres S.D. (fonctions générées: 30, 33, 39, 70 à 76). Il n'est pas possible de déclarer plus d'une S.D. d'utilisation M ou N par Programme. |
| | | E | Compte-rendu de contrôle : S.D. mouvement en sortie. La description comporte ou non les Rubriques facultatives. Dans le cas ou il n'y a pas de sélection de celles-ci, la description sera précédée des Rubriques ENPR et GRPR calculées lors de la génération et reprenant les vecteurs EN-PRR et GR-PRR. Dans le cas contraire, voir les commentaires de la zone 'DESCRIPTION SELECTIONNEE'. |
| | | I | Impression directe (ou par SYSOUT en OS IBM). La Structure commune éventuellement présente dans la description de l'Etat ne sera pas prise en compte. |
| | | J | Impression indirecte à traiter par un programme de |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | spool. Suppose l'utilisation de la Structure 00 pour définir une partie commune à toutes les lignes. |
| 17 | 2 | | <p>CORRESPONDANCE PRINCIPAL - RESULTAT</p> <p>Pour une S.D. principale (utilisation 'P'): indiquer le code dans le Programme de la S.D. résultat en sortie.</p> <p>Pour une S.D. résultat (utilisation 'R' ou 'D'): indiquer le code dans le Programme de la S.D. principale en entrée.</p> |
| 18 | 2 | | <p>CORRESPONDANCE SOURCE - SELECTIONNE</p> <p>Pour une S.D. mouvement (utilisation 'M' ou 'N'): indiquer le code dans le Programme de la S.D. compte-rendu de contrôle (utilisation E).</p> <p>Pour une S.D. compte-rendu de contrôle (utilisation 'E'): indiquer le code dans le Programme de la S.D. mouvement (utilisation 'M' ou 'N').</p> <p>Pour une S.D. sélectionnée (utilisation 'S'): indiquer le code dans le Programme de la S.D. source en entrée; sur la ligne d'appel de la S.D. source, indiquer le code dans le Programme de la S.D. sélectionnée.</p> |
| 19 | 1 | | <p>NIVEAU DE RUPTURE SUR MOUVEMENT</p> <p>Dans une S.D. mouvement, rang de la Rubrique d'indicatif qui permet de faire la distinction entre plusieurs mouvements relatifs à un même Segment de la S.D. permanente (ce niveau est donc nécessairement égal ou plus fin que le niveau de synchronisation de la S.D. mouvement).</p> <p>Tant qu'aucune rupture n'est détectée à ce niveau, les enregistrements traités appartiennent au même mouvement.</p> <p>Une rupture à ce niveau conditionne, par exemple, l'initialisation des variables d'erreurs associées au mouvement et le contrôle de correspondance avec la ou les S.D. principale(s).</p> <p>Par défaut il est égal au nombre de niveaux de rupture de la S.D.</p> |
| 20 | 4 | | <p>TYPE D'UNITE PHYSIQUE</p> <p>Généré dans la clause SELECT pour certains COBOLS :</p> <p>IBM DOS (variante 1) par exemple 2314, 3330, 2400...</p> <p>IBM 36 (variante 2) :</p> |

| NUM LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE .DK ou blanc = disque .PT = imprimante MICROFOCUS, COBOL II, IBM VISUAL SET (variante 3) : .EXT = génération de la clause EXTERNAL au niveau de la FD du fichier .LS = génération de la clause LINE SEQUENTIAL .EXLS = génération de la clause LINE SEQUENTIAL et de la clause EXTERNAL au niveau de la FD du fichier BULL DPS7 (variante 4) : .SSF = option WITH SSF dans la clause SELECT .OUT = option suffixe -SYSOUT après le nom du fichier dans la clause SELECT (WITH SSF est génééré). BULL DPS8 ASCII (variante 5) : .PT = imprimante .CR = lecteur de cartes .SSF = ORGANISATION IS GFRC SEQUENTIAL SSF CODE SET IS GBCD .IBM = ORGANISATION IS IBM-OS SEQUENTIAL .xxx = WITH xxx .un 'V' dans la quatrième position de cette zone permet de générer la clause 'VALUE OF ID'. La clause générée est : VALUE OF FILE-ID IS 3-XX00-IDENT où XX est le code structure de donnée programme. La zone 3-XX00-IDENT doit être définie en -W par l'utilisateur. Remarque : Le 'V' en quatrième position peut être combiné avec une des valeurs d'unité, sur la même ligne. BULL DPS8 BCD (variante 6) : .LIST = imprimante. .CARD = lecteur de cartes. Burroughs Large System, Unisys série A (variante 8) : .DK ou blanc = disque .DKS = disque de tri (avec ouverture T) .DKM = disque de fusion (avec ouverture T) .RD = lecteur .PT = imprimante .PO = fichier PORT .TP = bande .Pour les codes à 2 caractères le troisième peut préciser un ordre de fermeture particulier : |
|---------|------------------|---|
|---------|------------------|---|

| NUM LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE P = PURGE R = RELEASE L = LOCK S = SAVE .Un V en quatrième position fait générer la clause VALUE OF FILENAME IS 3-FF00-IDENT. Unisys 90/30 (variante 9) : .84 = disque avec nom externe .PT = imprimante .CR = lecteur de cartes .CP = perforateur de carte .TP = bande Prime (variante A) : .Blanc = PMFS .T7 = bande 7 pistes .T9 = bande 9 pistes .RD = lecteur .PR = imprimante .PU = perforateur .OP = imprimante off-line .TL = terminal Burroughs Medium System, Unisys série V (variante B) : .RD = lecteur de cartes. .DK = disque fixe. .DP = disque amovible. .TP = bande magnétique. .PT = imprimante. .PC = perforateur de cartes. .PP = perforateur de bande. .PR = lecteur de bandes perforées. .T7 = bande 7 pistes. .Le troisième caractère précise un ordre de fermeture particulier pour le fichier: P = PURGE R = RELEASE. Perkin-Elmer (variante J) : .blanc = disque .TP = bande .PR = imprimante .RD = lecteur .CA = cassette .PT = bande papier Siemens (variante T) par exemple 590, 432... Unisys 2200 (variante U) : .CR = lecteur de cartes .CP = perforateur de cartes .UN = UNISERVO |
|---------|------------------|--|
|---------|------------------|--|

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>.TP = bande</p> <p>.PN = imprimante avec nom externe. Si le champ 'complément du type d'unité physique' est renseigné, la clause RECORDING est également générée.</p> <p>.PT = imprimante sans nom externe</p> <p>.PF = imprimante avec nom externe et :</p> <p>VALUE OF PRINTER-FORMS 3-FF00-FORMS LINAGE IS 3-FF00-LINES TOP IS 3-FF00-TOP BOTTOM IS 3-FF00-BOTTOM</p> <p>Ces quatre noms-données sont à déclarer dans les zones de travail (-W) avec les valeurs appropriés.</p> <p>IBM 38 (variante Y) ou AS 400 (variante O) :</p> <p>.RD = lecteur</p> <p>.CP = perforateur de cartes</p> <p>.DB = base de données</p> <p>.PT = imprimante</p> <p>.TP = bande</p> <p>.DK ou blanc = disque</p> |
| 21 | 1 | | <p>COMPLEMENT AU TYPE D'UNITE PHYSIQUE</p> <p>IBM DOS (variante 1) :</p> <p>.R = lecteur</p> <p>.P = perforateur</p> <p>IBM 3-15 (variante 3) :</p> <p>.S = bande EBCDIC</p> <p>.C = bande ASCII</p> <p>BULL DPS8 ASCII (variante 5) :</p> <p>.S = CODE SET EBCDIC</p> <p>.C = CODE SET ASCII</p> <p>CDC COBOL 68 (variante E) :</p> <p>.S = RECORDING MODE IS EBCDIC</p> <p>UNISYS 2200 (variante U) :</p> <p>.S = RECORDING suivi du mode de blocage</p> <p>BULL DPS7 (variante 4) et DPS8 (variante 6) :</p> <p>.O = Si la valeur 'O' est saisie dans cette zone, l'option OPTIONAL n'est pas générée. Sinon, l'option OPTIONAL est générée par défaut.</p> <p>DEC VAX VMS (variante I) :</p> <p>.A = Ouverture de fichier avec l'option ALLOWING ALL et lecture séquentielle avec l'option REGARDLESS.</p> |
| 22 | 9 | | ARGUMENT, SELECTION ETATS ENR. |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE Cette zone a 3 rôles exclusifs les uns des autres : 1. Composition de l'argument d'une S.D. : ----- C'est la suite des Rubriques composant l'argument (ou indicatif) de la S.D., repérées par leur rang, tel qu'il est défini dans la description du Segment. Si cette zone n'est pas utilisée, l'indicatif principal (Rubriques de rang 1 à 9) sera repris par défaut. La sélection d'argument ne peut être précisée que sur la première ligne d'appel de la S.D. (zone SUITE à blanc) et concerne uniquement les Rubriques de la partie commune sélectionnée par cette ligne. 2. Sélection de Segments dans un fichier : ----- Se fait en mettant un '*' en première position, suivi des codes (sur deux caractères) des Segments à sélectionner (en plus de la partie commune). Permet, entre autres, d'extraire des tables dans une S.D. Pour sélectionner plus de 4 Segments , créer une ligne suite (voir la zone SUITE). Pour recomposer l'argument d'une S.D. et sélectionner des Segments, il faut indiquer la composition de l'argument sur la première ligne d'appel puis créer les lignes 'suite' nécessaires pour sélectionner les Segments; ceux-ci peuvent dépendre de plusieurs S.D. différentes (S.D. composite). Il ne faut pas, dans ce cas sélectionner explicitement la partie commune, la partie commune de la S.D. appelée sur la première ligne étant considérée comme la partie commune du fichier; les S.D. auxquels appartiennent les autres Segments ne doivent pas avoir de partie commune. Il est préférable, pour créer une S.D. composite, de sélectionner un Segment unique qui regroupera les Segments choisis par appel de Segment (voir le Chapitre "Segments", sous-chapitre "Description" du Manuel de Référence DICTIONNAIRE. 3. Sélection d'Etats dans un support : ----- C'est la suite des codes Etats (sur un caractère) à éditer dans ce Programme. Si la zone n'est pas renseignée, tous les Etats du support seront édités. |
|-----|-----|------------------|--|
|-----|-----|------------------|--|

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|---|--|
| | | | <p>Pour sélectionner plus de 9 Etats, créer une ligne 'suite' (voir la zone SUITE).</p> <p>Modification des codes de Segments et d'Etats : -----</p> <p>Il est possible de changer pour un Programme, le code d'un Segment ou le code d'un Etat, en écrivant une ligne d'appel par Segment ou Etat à renommer et en indiquant dans l'argument le code en Bibliothèque suivi du code dans le Programme, reliés par le signe '=', par exemple :</p> <p>1=2 changement du code Etat 1 en code Etat 2. *01=02 changement du code Segment 01 en code Segment 02.</p> |
| 23 | 1 | E I S | <p>CHOIX DU TYPE DE FORMAT</p> <p>Le choix du type de format est réservé aux S.D. d'utilisation autre que I ou J.</p> <p>Format d'entrée (option par défaut pour les fichiers d'utilisation : M, N, E).</p> <p>Format interne (option par défaut pour les fichiers d'utilisation autre que : M, N, E).</p> <p>Format de sortie</p> <p>Remarque : les Rubriques constituant les Segments ne doivent pas dépasser 999 caractères.</p> |
| 24 | 1 | BLANC V W E U | <p>DESCRIPTION SELECTIONNEE</p> <p>Précise si les Rubriques facultatives (ENPR, GRPR, ERUT) figurant dans la description de la S.D. sont à reprendre :</p> <p>BLANC Pas de prise en compte des Rubriques facultatives.</p> <p>V Prise en compte de toutes les Rubriques facultatives.</p> <p>W Identique à la valeur 'V', mais alors la Rubrique ENPR représente le vecteur erreur réduit. (Réservé aux S.D. d'utilisation M, N ou E).</p> <p>E Prise en compte des Rubriques ENPR et GRPR seules.</p> <p>U Prise en compte de la Rubrique ERUT seule.</p> <p>Dans une S.D. mouvement (utilisation 'M' 'N' ou 'E'), ces Rubriques doivent être obligatoirement en tête de la description et permettent de passer les résultats du contrôle dans la mise à jour.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | 1..9,0 | <p>. ENPR : n + 1 positions pour les valeurs V ou E, m + 1 positions pour la valeur W (n est le nombre de Rubriques élémentaires sur l'ensemble de la description de la S.D., m est le nombre maximum de Rubriques élémentaires d'un segment de la S.D.)</p> <p>La position supplémentaire est l'erreur d'identification. Elle initialise le vecteur EN-PRR.</p> <p>. GRPR (1 position par segment + 1 pour l'erreur de groupe), initialise le vecteur GR-PRR.</p> <p>Dans une S.D. d'utilisation différente, elles peuvent figurer n'importe où et sont de format libre.</p> <p>Pour le module TABLES, indique le numéro de sous-schéma désiré (cf. manuel de référence TABLES). Pour un fichier d'utilisation SQL, indique le numéro de sous-schéma désiré (sélection d'une colonne d'une Table).</p> |
| 25 | 1 | BLANC | <p>TYPE DE DESCRIPTION GENEREE</p> <p>Cette option permet d'obtenir plusieurs modes d'agencement des enregistrements d'une S.D. à partir des mêmes descriptions en Bibliothèque.</p> <p>Enregistrements redéfinis (option par défaut).</p> <p>1 Enregistrements à la suite (partie commune suivie des différentes parties spécifiques), sans valeur initiale ni répétition d'enregistrements.</p> <p>Si la description de la S.D. apparaît en FILE SECTION COBOL, la zone NOMBRE NIVEAU DU SEGMENT doit avoir la valeur 2.</p> <p>2 Enregistrements à la suite avec prise en compte des valeurs initiales spécifiées sur les lignes d'appel des Rubriques dans les Segments, ou à défaut initialisation à blanc ou zéro selon le format. Option réservée aux S.D. en organisation W ou L.</p> <p>3 Enregistrements à la suite avec prise en compte du nombre de répétitions spécifié dans la définition des Segments, sans valeurs initiales.</p> <p>Si la description de la S.D. apparaît en FILE SECTION COBOL, la rubrique NOMBRE NIVEAU DU SEGMENT doit avoir la valeur 2.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>Ce type de description ne peut être employé pour une S.D. ayant un nombre de répétitions sur la partie commune (utiliser l'organisation W associée à l'utilisation T).</p> |
| | | 4 | <p>Enregistrements à la suite avec prise en compte du nombre de répétitions spécifié dans la définition des segments. La zone NOMBRE NIVEAU DU SEGMENT doit avoir la valeur 3.</p> <p>Remarque spécifique au module DIALOGUE : En cas de type de description '4' et de nombre niveau '3', l'indice n'est pas généré.</p> <p>Le niveau 2 permet d'accéder à la table constituée par la répétition d'un même enregistrement (FFEET).</p> <p>Le niveau 1 regroupe l'ensemble des informations de la S.D. (partie commune et parties spécifiques répétées ou non).</p> <p>Ce type de description ne peut être employé pour une S.D. ayant un nombre de répétitions sur la partie commune (utiliser l'organisation W associée à l'utilisation T).</p> |
| | | 6 | <p>Seulement pour l'interface GIP. Le nombre de niveaux est le même que pour le type de description 4.</p> |
| | | 7 | <p>Description étalée d'un dossier conforme aux spécifications de GIP version 7.</p> <p>Même description que pour le type 6.</p> |
| | | 8 | <p>Description étalée d'un dossier conforme aux spécifications de GIP version 8.</p> <p>Même description que pour le type 7 sauf que la partie 'SYSTEME' est plus longue de 24 caractères.</p> <p>Pour ces trois derniers types, il est possible de décrire la partie commune au niveau du dossier en utilisant le type de description 4 si la partie commune générée ne convient pas.</p> |
| 26 | 1 | | <p>NOMBRE NIVEAU DU SEGMENT</p> <p>Cette option, employée en relation avec la précédente, définit le niveau COBOL des descriptions de S.D., de</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE Segments ou de Rubriques. |
|-----|-----|------------------|---|
| | | 1 | <p>Dans les descriptions ci-après on entend par niveau S.D., le niveau COBOL de la zone DD00 (éventuellement 1-DD00, 2-DD00).</p> <p>Niveau 01 pour S.D. et Segments (valeur par défaut).</p> <p>Si la description de la S.D. apparaît en FILE SECTION COBOL, les Segments doivent obligatoirement être redéfinis.</p> <p>Si la S.D. est sans partie commune avec une description étalée, le niveau S.D. n'apparaît pas, sauf quand le type de description est à blanc.</p> |
| | | 2 | <p>Niveau 01 pour S.D. et 02 pour Segments. Si le type de description est à blanc, la S.D. apparaît, comme les Segments, en niveau 2. (utiliser de préférence l'organisation 'L' et définir les nombres niveaux supérieurs par ligne '-W').</p> |
| | | 3 | <p>Niveau 02 pour S.D. et 03 pour Segments, lorsqu'il est associé aux types de description 1, 2 ou 3,</p> <p>Niveau 01 pour S.D. et 03 pour Segments lorsqu'il est associé aux types de description 4 ou 6,</p> <p>Niveau 03 pour S.D. et Segments lorsqu'il est associé au type de description blanc.</p> <p>Réservé aux S.D. en organisation W ou L.</p> |
| | | 4 | <p>Niveau 02 pour les rubriques de premier niveau. Les rubriques élémentaires appartenant à une rubrique groupe apparaissent mais les niveaux S.D et Segments disparaissent.</p> <p>Réservé aux S.D. d'organisation 'L' et utilisation 'D' Le niveau 01 est à définir par ligne '-W'.</p> |
| | | 5 | <p>Niveau 01 pour les rubriques de premier niveau. Les rubriques élémentaires appartenant à une rubrique groupe apparaissent mais les niveaux S.D et Segments disparaissent.</p> <p>Réservé aux S.D. en organisation 'L' ou 'W' et utilisation 'D'.</p> |
| | | 6 | <p>Niveau 02 pour les rubriques de premier niveau. Les rubriques élémentaires appartenant à une rubrique groupe, les niveaux S.D et Segments disparaissent.</p> <p>Réservé aux S.D. d'organisation 'L' et utilisation 'D'</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|----------------------------------|--|
| | | 7 | <p>Le niveau 01 est à définir par ligne '-W'. Valable uniquement pour un Ecran standard Dialogue.</p> <p>Niveau 01 pour les rubriques de premier niveau. Les rubriques élémentaires appartenant à une rubrique groupe, les niveaux S.D et Segments disparaissent.</p> <p>Réservé aux S.D. en organisation 'L' ou 'W' et utilisation 'D'. Valable uniquement pour un Ecran standard Dialogue.</p> |
| 27 | 2 | 00 ALPHABET. ALPHANUM. | <p>EMPLACEMENT DE LA DESCRIPTION</p> <p>ZONE PSEUDO-NUMERIQUE, blancs et zéros équivalents.</p> <p>Ce champ ne concerne que la description principale de la S.D. (DDSS), et non les descriptions précédées d'un radical (1-DDSS OU 2-DDSS).</p> <p>Il permet d'obtenir une description de S.D. dans une zone particulière (zone de communication avec les gestionnaires de bases de données, LINKAGE SECTION, que l'utilisateur doit définir par ligne '-W'), ou en tête de la WORKING-STORAGE SECTION.</p> <p>Cette option est réservée aux S.D. en organisation 'L', 'D' ou 'W' et doit être généralement combinée avec les deux précédentes.</p> <p>Pour effectuer une description de structure de données en WORKING-STORAGE, il est préférable d'appeler la S.D. par l'écran 'zones de travail et de liens' (-W). (Cf. Manuel de Référence LANGAGE STRUCTURE).</p> <p>La description de la S.D. est insérée après toutes les lignes '-W' (valeur par défaut).</p> <p>La description de la S.D. est insérée après toutes les lignes '-W' dont le numéro de ligne à 5 positions commence par cette valeur. Les lignes '-W' et la description se trouvent, dans le programme généré, en tête de WORKING-STORAGE SECTION avant les descriptions des S.D. d'organisation 'W' et dont le code dans le Programme est supérieur à cette valeur (ne pas prendre un code emplacement identique au code dans le Programme d'une S.D. d'organisation 'W').</p> <p>La description de la S.D. est insérée après toutes les lignes '-W' dont le numéro de ligne à 5 positions commence par cette valeur. Les lignes '-W' et la description se trouvent, dans le programme généré, en fin de WORKING-STORAGE SECTION parmi les zones-utilisateur.</p> <p>L'emplacement indiqué sur la ligne définissant la S.D.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>(zone SUITE à blanc), est reporté par défaut sur toutes ses lignes suite, au niveau de la documentation et de la génération.</p> <p>Cependant il est possible d'attribuer des emplacements différents à chaque description de Segment d'une S.D. dans le Programme en multipliant les lignes d'appel pour cette S.D. et en précisant pour chacune une sélection de Segments et un emplacement. La S.D. doit alors avoir une description détaillée.</p> <p>ATTENTION : pour ORACLE, vous devez utiliser des valeurs numériques pour que la DECLARE SECTION soit générée correctement (avec les zones données et indicateurs).</p> |
| 28 | 10 | | <p>ZONE COMPLEMENTAIRE</p> <p>Cette zone est de la forme DDSSRRRRRR avec :</p> <ul style="list-style-type: none"> . DD = code structure de données, . SS = code segment . RRRRRR = code rubrique. <p>Elle a trois significations :</p> <ul style="list-style-type: none"> . NOMINAL, SYMBOLIC ou ACTUAL KEY suivant la variante, sur les matériels sur lesquels elle est nécessaire, sauf pour H66 BCD et les fichiers en organisation V. <p>L'utilisateur doit définir la zone de WORKING correspondante: 1-DDSS-RRRRRR.</p> <p>Le positionnement de cette clé, ainsi que la lecture du fichier sont à programmer en langage structuré.</p> <ul style="list-style-type: none"> . Identification du fichier pour CII-HB H66 BCD (variante 6). La clause correspondante 'VALUE OF' ne sera générée que si elle est renseignée. . Zone code-retour des opérations d'entrée-sortie pour les fichiers en organisation V. La clause correspondante 'FILE STATUS IS' ne sera générée que si elle est renseignée. |
| 29 | 6 | | <p>CODE DE LA CLE DANS L'ENREGISTREMENT</p> <p>Clé dans l'enregistrement, obligatoire pour les fichiers en organisation indexée.</p> |
| 30 | 6 | | <p>NOM DU CODE ENREGISTREMENT</p> <p>C'est le code mnémonique de la Rubrique permettant d'identifier par ses valeurs les différents types d'enregistrements du fichier.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>Il se trouve obligatoirement dans la partie commune (code enregistrement 00).</p> <p>Ce code peut également être précisé au niveau de la définition du Segment partie commune, dans la zone NOM DU CODE STRUCTURE et sert alors de valeur par défaut au niveau de la génération.</p> |

4.2. MODIFICATION DU DEBUT DE PROGRAMME (-B)

MODIFICATION DE DEBUT DE PROGRAMME (-B)

L'écran '-B' a pour objet de compléter ou de modifier le début des Programmes générés (de 'IDENTIFICATION DIVISION' jusqu'à la ligne 'FILE SECTION' incluse).

Son emploi est exceptionnel sur la plupart des matériels.

Par exemple :

Clause SELECT pour les fichiers en accès relatif.

Cas CII-HB H64 : clause COPY SELECT et COPY FD dans une TPR TDS.

SUPPRESSION DE LIGNES GENEREES

L'existence d'une ligne '-B' portant le code d'une section seule ou d'un paragraphe entraîne le remplacement des lignes générées automatiquement pour cette section ou ce paragraphe.

Les lignes '-B' sont ignorées dans un programme issu de REVERSE ENGINEERING.

```
-----  
!                                     !  
!           DOCUMENTATION           ! P0*JFL.LURE.TES.972 !  
!                                     !  
! MODIFICATION DEBUT DU PROGRAMME 1 P 2 TES001 test 001 !  
!                                     !  
! 3 4 5 6 7                         !  
! A SE PA NLG INSTRUCTION A INSERER !  
! * 60 000 DATA DIVISION           !  
! * 70 000 SUB SCHEMA SECTION       !  
! * 80 000 FILE SECTION             !  
! 99 99 000 SUPPRIMEE               !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
! O: C1 CH: Ptes001 B               !  
!                                     !  
-----
```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| 1 | 1 | | TYPE D'ENTITE Cette zone est saisissable, ce qui permet de copier des lignes d'un programme vers un écran et vice versa. |
| | | O | Ecran. |
| | | P | Programme. |
| 2 | 6 | | CODE PROGRAMME OU ECRAN |
| 3 | 1 | | CODE ACTION |
| 4 | 2 | | SECTION GENEREE Dans une macro-structure, la section générée peut être paramétrée. |
| 5 | 2 | | PARAGRAPHE GENERE Les codes section ET paragraphe utilisables dans les principaux cas sont les suivants: |
| | | 05 | Commentaire PAF. |
| | | 10 | PROGRAM-ID. |
| | | 20 | AUTHOR. |
| | | 30 | DATE-COMPILED. |
| | | 40 | ENVIRONMENT DIVISION. |
| | | 00 | CONFIGURATION SECTION. |
| | | 00 00 | SOURCE-COMPUTER. |
| | | 00 10 | OBJECT-COMPUTER. |
| | | 00 20 | SPECIAL-NAMES. |
| | | 01 | INPUT-OUTPUT SECTION. |
| | | 01 | FILE-CONTROL. |
| | | 01 FF | SELECT FF-FICHER |
| | | 01 00 | I-O-CONTROL. Ce paragraphe n'est pas généré automatiquement sauf, dans certains cas, en variante 6 (H66 BCD). |
| | | 01 90 | RECEIVE-CONTROL (TANDEM) |
| | | 60 | Est réservé à la ligne 'DATA DIVISION'. |
| | | 99 99 | Est réservé à la ligne 'FILE SECTION'. |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | 9* FF | <p>Les clauses FD générées ne peuvent être ni modifiées ni annulées.</p> <p>Dans une macro-structure, le paragraphe généré peut être paramétré.</p> <p>Réécriture de la clause FD pour le fichier FF.</p> |
| 6 | 3 | \$n | <p>NUMERO DE LIGNE</p> <p>ZONE NUMERIQUE PARAMETRABLE</p> <p>Il est recommandé de numéroter les lignes de 10 en 10 pour permettre d'en insérer de nouvelles par la suite.</p> <p>Dans une Macro-Structure, il est possible de paramétrer les deux premiers caractères du numéro de ligne.</p> |
| 7 | 66 | S | <p>INSTRUCTION A INSERER</p> <p>Si l'instruction doit être en marge 'A' du COBOL, elle doit commencer en deuxième colonne de ce champ. Si elle doit être en marge 'B', elle doit commencer à la sixième colonne de ce champ. Sinon elle sera cadrée en colonne 7 du COBOL (commentaire ou suite de ligne).</p> <p>(ou tout mot commençant par 'S') : Entraîne la suppression de la section ou du paragraphe générés.</p> <p>Dans une macro-structure, cette zone peut être paramétrée.</p> |

4.3. ZONES DE TRAVAIL ET DE LIEN (-W)

ZONES DE TRAVAIL ET DE LIEN (-W)

L'écran ZONES DE TRAVAIL ET DE LIEN permet :

- . l'appel de Structures de Données,
- . l'appel direct de Rubriques (sans passer par la définition d'un Segment), avec prise en compte du format interne, d'entrée ou de sortie,
- . l'appel de Rubriques qui n'existent pas encore dans la Base,
- . l'écriture de langages source autre que le COBOL dans des Programmes de nature 'S',
- . tout type de déclaration directement en COBOL.

L'emploi de cette possibilité doit être limité à la déclaration de clauses non générées automatiquement par PACBASE, telles que le titre 'LINKAGE SECTION'.

- . la génération d'indices utilisés lors de la recherche dans une table (avec l'opérateur SCH). Ceci est réalisé en associant une dimension de table (clause OCCURS) au couple Structure de Données et Code Segment indiqués dans la zone Description.

CONSEILS D'UTILISATION

L'écran d'appel des Structures de Données (-CD) définit les ressources externes au Programme (fichier, base de données...); les zones de travail et de lien sont regroupées dans les '-W'. On peut ainsi les ordonner facilement.

De plus, les lignes '-W' d'une Macro-Structure sont incorporées aux programmes appelants, et non les appels de Structure de Données (-CD). Il faut néanmoins faire attention aux indicatifs des '-W' de la Macro-Structure, afin qu'ils ne rentrent pas en conflit avec ceux du Programme appelant ou d'autres Macro-Structures.

APPEL DE STRUCTURES DE DONNEES

Il est fait par des lignes de type 'F'. Un guide de saisie permet de rentrer les caractéristiques de la Structure de données. (voir la zone 'Type de déclaration' dans la description de l'écran).

```
-----  
!                                     !  
!           DOCUMENTATION           ! P0*JFL.LURE.TES.972 !  
!                                     !  
! ZONES DE TRAVAIL DU PROGRAMME 1 P 2 TES001 test 001 !  
!                                     !  
! DEBUT DU NUMERO DE LIGNE : BB 3 !  
! 4 5 6 7 8 9 !  
! A NLG S NIVEAU DESCRIPTION TABL !  
! 100 01 MEMENR. !  
! 120 I 05 I-ZZ00-CP1 !  
! 140 VALUE '00'. !  
! * 200 F SD: XW BIB: XW SEL: 02_____ FORM: I DESC: 2 NIV: 1 ORG: _ SS: !  
! 300 F SD: XW BIB: XW SEL: 04_____ FORM: I DESC: 2 NIV: 1 ORG: _ SS: !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
! O: C1 CH: -W !  
!                                     !  
-----
```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| 1 | 1 | | <p>TYPE D'ENTITE</p> <p>Cette zone est saisissable, ce qui permet de copier des lignes d'un programme vers un écran et vice versa.</p> <p>O Ecran.</p> <p>P Programme.</p> |
| 2 | 6 | | CODE PROGRAMME OU ECRAN (OBLIGATOIRE) |
| 3 | 2 | | <p>DEBUT DU NUMERO DE LIGNE (OBLIGATOIRE)</p> <p>ZONE PSEUDO-NUMERIQUE</p> <p>Dans cette zone, vous pouvez saisir :</p> <ul style="list-style-type: none"> - des caractères alphabétiques (les blancs seront remplacés par des zéros); - des caractères numériques; - \$n pour indiquer une valeur paramétrée d'une MSP. <p>Le contenu de la zone sert à déterminer l'emplacement et la séquence selon lesquels les données décrites dans cet écran seront générées en DATA DIVISION. Le numéro de séquence (5 car.) d'une ligne de description de données est formé par concaténation du contenu de cette zone et du NUMERO DE LIGNE de la ligne.</p> <p>Pour un programme Batch :</p> <p>AA à ZZ 0A à 0Z</p> <p>Si le DEBUT DU NUMERO DE LIGNE est inférieur à '00' (ce qui inclut aussi les valeurs de '0A' à '0Z'), les données décrites dans l'écran seront générées en début de WORKING-STORAGE SECTION. Par rapport aux Structures de Données appelées via l'écran Appel de S.D. (-CD), elles seront générées :</p> <ul style="list-style-type: none"> - avant la description des S.D. d'ORGANISATION 'W', et ayant un CODE S.D. DANS LE PROGRAMME supérieur ou égal au contenu de cette zone; - avant la description des S.D. d'ORGANISATION 'L' ou 'D', et ayant un EMPLACEMENT DE LA DESCRIPTION supérieur ou égal au contenu de cette zone. <p>00 à 09 1A à 19 ... 9A à 99</p> <p>Si le DEBUT DU NUMERO DE LIGNE est supérieur à '00', les données décrites dans l'écran seront générées en WORKING-STORAGE SECTION, après toutes les Structures de Données dont l'EMPLACEMENT DE LA DESCRIPTION (écran -CD) est inférieur au contenu de cette zone.</p> <p>Pour un programme T.P. :</p> <p>AA à 0Z</p> <p>Si le DEBUT DU NUMERO DE LIGNE est inférieur à '00'</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE (ce qui inclut aussi les valeurs de '0A' à '0Z'), les données décrites dans l'écran seront générées en WORKING-STORAGE SECTION. |
|-----|-----|--|--|
| | | 00 à 99 | Sinon, elles seront générées en LINKAGE SECTION. |
| | | AA | Cette valeur est utilisée par le système pour des données générées automatiquement. |
| | | 00 | Cette valeur est utilisée par le système pour des données générées automatiquement. |
| | | | Selon la variante de génération, d'autres valeurs peu- vent être réservées pour des usages particuliers. |
| | | 99 | Avec un NUMERO DE LIGNE = '999' : cette valeur est utilisée par le système pour l'instruction 'PROCEDURE DIVISION'. Vous pouvez l'utiliser afin de créer une ligne de numéro de séquence '99999', qui remplacera la ligne 'PROCEDURE DIVISION' générée automatiquement. |
| | | \$n | Dans une Macro-Structure Paramétrée, vous pouvez para- métriser le contenu de cette zone. |
| 4 | 1 | | CODE ACTION |
| 5 | 3 | | FIN DU NUMERO DE LIGNE ZONE PSEUDO-NUMERIQUE Il est recommandé de numéroter les lignes de 10 en 10 pour permettre d'en insérer de nouvelles par la suite. Dans une Macro-Structure, la fin du numéro de ligne peut être paramétrée. |
| 6 | 1 | blanc - * \$ F | TYPE DE DECLARATION Les données saisies dans les zones NIVEAU DE DECLARATION et DECLARATION DES DONNEES DE TRAVAIL sont générées telles qu'elles sont saisies. Continuation d'un littéral. Commentaire (dimension de table exclue), inséré dans le programme généré (COBOL ANS seulement). Cette valeur figurera dans le généré COBOL en colonne 7 et le reste de la ligne working sera repris tel quel. Appel d'une Structure de Données. (Voir le sous-chapitre "Appel des Structures de Don- nées" pour une description détaillée des informations) |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>Si la ligne est créée à blanc elle sera affichée avec un guide de saisie des informations nécessaires. Celles-ci correspondent exactement aux informations équivalentes de la ligne d'appel de Structures de Données pour les organisations 'W' et 'L':</p> <ul style="list-style-type: none"> . code S.D. dans le Programme (paramétrable dans une Macro-Structure), . code de la S.D. en bibliothèque (paramétrable dans une Macro-Structure), . sélection d'enregistrements (non précédés d'une '*'), (paramétrable dans une Macro-Structure), (dans un écran il n'est pas possible de renommer un segment), . choix du type de format (I, E ou S), . type de description généré (1 à 8), . nombre niveau de l'enregistrement (1 à 5), . organisation, . sous-schéma. <p>Pour batch uniquement, les lignes '-W' de type 'F' sont traitées comme des lignes d'appel de S.D. (-CD). Si deux lignes '-W' de type 'F' concernant la même S.D. (même code S.D. dans le programme) sont séparées, elles seront quand même générées l'une derrière l'autre.</p> |
| | | I | Appel de Rubrique avec prise en compte de son format interne. |
| | | E | Appel de Rubrique avec prise en compte de son format d'entrée. |
| | | S | Appel de Rubrique avec prise en compte de son format de sortie. |
| | | | <p>Pour ces trois types d'appels, un contrôle sur l'existence au Dictionnaire est effectué. Un chaînage particulier est réalisé, il interdit l'annulation de cette Rubrique si les lignes où elle est appelée n'ont pas été préalablement annulées.</p> <p>Si la Rubrique n'existe pas dans le Dictionnaire, un "warning" est envoyé par le Système.</p> <p>Dans le cas d'une substitution globale (.C2), le contrôle de la Rubrique ne sera pas fait mais en revanche les chaînages seront bien créés.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>Pour ces trois derniers types de déclaration, le nom-donnée déclaré doit être du type W-FFEE-RRRRRR avec :</p> <ul style="list-style-type: none"> . W un préfixe quelconque (par exemple I, E ou S), . FFEE un code Segment quelconque, . RRRRRR un code Rubrique existant au Dictionnaire. <p>Ce nom-donnée doit être cadré à gauche de la zone Déclaration.</p> <p>MODULE DIALOGUE (IMS) -----</p> <p>M Déclaration d'un Ecran comme moniteur.</p> <p>C Affectation d'un Ecran à un autre Ecran déclaré comme moniteur (type 'M').</p> |
| 7 | 17 | | <p>NIVEAU DE LA DECLARATION</p> <p>Nombre niveau (01, 05, ...) ou nom de section (LINKAGE SECTION).</p> <p>Dans une Macro-Structure, le niveau de la déclaration est paramétrable.</p> |
| 8 | 48 | | <p>DECLARATION DES DONNEES DE TRAVAIL</p> <p>Dans tous les cas, on doit choisir les noms de zones (nom-données) en utilisant la normalisation PACBASE. La structure de ces noms est W-FFEE-RRRRRR, avec:</p> <ul style="list-style-type: none"> .W = code zone de travail (chiffre ou lettre), .FF = code de la S.D. dont la zone est extraite, .EE = code enregistrement, .RRRRRR = code de la Rubrique. <p>Si cette normalisation est respectée, les références croisées Rubrique/Programme sont établies automatiquement.</p> <p>Dans le programme généré, cette zone suit immédiatement la précédente.</p> <p>Dans une Macro-Structure, la déclaration de la donnée peut être paramétrée.</p> <p>REMARQUE : Lors de l'appel d'une Rubrique dans une zone de travail et de lien, et dans le cas où le code Rubrique existe au dictionnaire, il est obligatoire de saisir le nom-donnée dans cette zone. Dans le cas</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE contraire, le généré obtenu sera de la forme: 03 FFEE-CORUB PICTURE X. |
|-----|-----|------------------|--|
| 9 | 5 | | DIMENSION TABLE ZONE NUMERIQUE PURE Déclenche la génération de trois indices : .IFFEEM initialisé à la valeur mentionnée, .IFFEEL initialisé à zéro, utilisable pour le rem- plissage de la table, donne le nombre de postes effectivement chargés. .IFFEER initialisé à zéro, utilisable pour la re- cherche en table. La donnée déclarée doit être de la forme : W-FFEE ou W-FFEE-RRRRRR, et cadrée à gauche. Dans une Macro-Structure, la dimension de la table peut être paramétrée. Cette zone n'est pas prise en compte dans les lignes de type 'F'. |

5. LANGAGE STRUCTURE

5.1. DESCRIPTION DES TRAITEMENTS (-P)

DESCRIPTION DES TRAITEMENTS

L'écran de description des traitements permet l'écriture de tous les types de traitements.

Ceux-ci sont structurés en fonctions et sous-fonctions. Une fonction ou sous-fonction est un ensemble d'instructions exécutées sous un même conditionnement (condition simple, boucle...) qui représente le type de fonction ou sous-fonction.

Chaque fonction/sous-fonction possède un type et un niveau; ce niveau établit la hiérarchie des sous-fonctions entre elles.

Les instructions sont exprimées par un opérateur suivi d'opérandes. L'opérateur est indiqué sur la première ligne, les opérandes peuvent occuper plusieurs lignes.

NIVEAU DES SOUS-FONCTIONS

Les fonctions ont toujours le niveau 05. Les sous-fonctions ont un niveau 10 par défaut. Elle peuvent prendre les niveaux 06 à 98.

Le niveau 99 sert à conditionner une séquence d'instructions à l'intérieur d'une sous-fonction.

Dans une fonction, une sous-fonction de niveau 15 fait partie de la sous-fonction de niveau 06 à 14 qui la précède. On parlera d'une sous-fonction de niveau inférieur si son nombre niveau est plus grand.

Une sous-fonction incluse dans une autre n'est évidemment exécutée que sous la condition d'exécution de la fonction supérieure.

TRAITEMENTS ELEMENTAIRES

Le niveau 99 est réservé à l'écriture de traitements élémentaires à l'intérieur d'une sous-fonction. Il permet d'écrire une condition sans changer de code sous-fonction. La condition s'applique jusqu'au niveau suivant.

TYPES DE STRUCTURES

Une fonction ne peut être que du type 'IF... THEN...' ('IT') si elle est soumise à une condition d'exécution, ou du type 'BLOCK' ('BL') sinon. Ce type est indiqué dans la zone TYPE, sur sa première ligne. PACBASE prend par défaut l'une de ces options selon que la zone CONDITION est renseignée ou non.

Pour les sous-fonctions et les traitements élémentaires les options par défaut sont les mêmes, mais on peut en outre, indiquer des types de structure plus complexes.

ALTERNATIVE

Une sous-fonction de type 'IT' peut être suivie d'une sous-fonction de même niveau de type 'ELSE' ('EL'). Celle-ci sera exécutée si la première ne l'a pas été. Il ne doit pas y avoir d'autre sous-fonction de niveau égal ou supérieur entre les deux sous-fonctions 'IT' et 'EL'.

AIGUILLAGE MULTIBRANCHE

Les aiguillages sont réalisés en indiquant sur une première sous-fonction le type de structure 'CASE OF' ('CO') et comme conditionnement le nom de la variable qui conditionne les différents traitements.

Les traitements seront décrits à la suite par des sous-fonctions de type 'IT' et de niveau immédiatement inférieur à celui du 'CO', avec comme condition la valeur de la variable correspondant à chaque traitement.

Les sous-fonctions 'IT' doivent toutes être de même niveau. Elles peuvent être décomposées en sous-fonctions de niveau inférieur. La dernière sous-fonction du 'CO', de même niveau que les 'IT', peut être de type 'BL' (non conditionnée): elle sera exécutée si les sous-fonctions précédentes ne l'ont pas été.

Le 'CO' est interrompu par la première sous-fonction de niveau supérieur aux 'IT'.

EXEMPLE :

```
15      CO  XXNN-CORUB
16      IT  valeur1
16      IT  valeur2
16      IT  valeur3
16      BL
```

BOUCLES

Il existe trois types de structures correspondant à des boucles DO WHILE ('DW'), DO UNTIL ('DU') et DO ('DO').

Une sous-fonction de type 'DW' n'est exécutée que si la condition indiquée est réalisée et tant qu'elle le reste.

Une sous-fonction de type 'DU' est exécutée au moins une fois et jusqu'à ce que la condition indiquée soit réalisée.

Une sous-fonction de type 'DO' est exécutée autant de fois que précisé dans la condition.

Pour les deux premiers types il faut veiller à faire évoluer les variables du conditionnement dans la sous-fonction pour éviter de boucler indéfiniment.

MESSAGE D'ERREUR DE TYPE 'WARNING'

A l'affichage d'un message de type 'WARNING', le caractère 'W' apparaît dans la zone CODE ACTION. L'utilisateur a la possibilité d'outrepasser le message en transmettant une nouvelle fois.

CONDITION D'EXECUTION

Le Langage Structuré sépare la condition d'exécution d'un traitement de son écriture.

L'écriture de la condition d'exécution commence sur la première ligne d'une (sous-)fonction et peut se continuer sur plusieurs lignes.

Ces lignes peuvent comporter ou non des traitements, ceux-ci seront, de toute façon, exécutés sous la condition globale.

Note sur les opérateurs de traitement de date du Module DIALOGUE :

Lorsque la condition est écrite sur plusieurs lignes, les lignes suite ne doivent pas comprendre d'opérandes.

Pour faciliter l'écriture de la condition, la zone TYPE peut être utilisée pour indiquer les relations 'et' ('AN') et 'ou' ('OR'). Les parenthèses doivent être indiquées. Il est souhaitable, pour une bonne lisibilité du Programme, de faire figurer toutes les relations logiques ET et OU dans la zone TYPE.

TRAITEMENTS

Les traitements sont décrits par des opérateurs suivis d'opérandes.

Ceci permet une bonne lisibilité du Programme en isolant les 'verbes' des données manipulées. Les opérateurs sont traduits en COBOL en tenant compte des renseignements fournis pour les fichiers et des particularités de chaque compilateur.

Un opérateur n'est indiqué qu'une fois même si les opérandes continuent sur plusieurs lignes. Il y a une exception pour l'opérateur '*' qui doit être répété sur chaque ligne de commentaires.

DEBRANCHEMENTS

La structure du Programme doit rester linéaire. On ne saute d'une fonction à une autre qu'en séquence. On ne se débranche jamais à une fonction ou sous-fonction précédente. Le seul type de débranchement légitime est donc celui qui permet d'abandonner un traitement en cours pour aller à la fin de la fonction ou de la sous-fonction dans laquelle on se trouve. Des opérateurs spécifiques permettent d'obtenir tous les débranchements de ce type.

Certains opérateurs et types de fonctions décrits ne sont utilisables qu'avec le module DIALOGUE.

CADRAGES DE ZONE

En TP, les zones OPERANDES et CONDITION D'EXECUTION ne sont pas affichées en totalité. Il est toutefois possible de faire afficher ces zones cadrées à droite, faisant ainsi apparaître les quatre derniers caractères de la première et les huit derniers de la seconde. Aucune mise à jour n'est alors possible.

Pour chaque zone, quatre caractères masqués correspondent à l'allongement éventuel de la zone résultant du remplacement de paramètres.

Pour la zone CONDITION, les quatre caractères supplémentaires sont accessibles en batch mais non en TP.

ATTENTION : si une ligne comporte une condition utilisant ces quatre caractères, la mise à jour de cette zone en TP les fera disparaître.

VISUALISATION DES TRAITEMENTS GENERES

Un écran permet d'écrire des traitements spécifiques tout en visualisant les titres des traitements générés.

Cet écran est obtenu grâce au choix :

CH: PaaaaaPG

Les traitements sont décrits dans les mêmes conditions que sur l'écran '-P'.

RECOMMANDATION

L'affichage de cet écran mettant en oeuvre plusieurs programmes et utilisant une place mémoire non négligeable, il est recommandé de l'utiliser uniquement lorsqu'il est nécessaire de visualiser les titres des traitements générés. Bien que cela puisse être tentant, l'usage de l'écran '-PG' ne doit pas se substituer à l'utilisation courante de l'écran '-P'.

DEBRANCHEMENT SUR L'ECRAN 'TC'

L'utilisateur peut se débrancher sur l'écran 'TC' (se référer au sous-chapitre "TITRES ET CONDITIONS DES TRAITEMENTS" chapitre "TITRES, CONDITIONS ET NIVEAUX DES TRAITEMENTS") en utilisant la touche fonction déterminée à cet effet (en standard PF9).

```
-----  
!  
!          DOCUMENTATION                                P0*JFL.LURE.TES.972 !  
! TRAITEMENTS PROGRAMME 1 P 2 TES001 test 001          FONCTION: 3 OA !  
!                                     4                                     5 !  
! A SS NLG OPE OPERANDE                                NVTY CONDITION !  
! 6 7 8 9 10                                         1112 13 !  
! * BA N DEBUT DE PROGRAMME                          10BL !  
! * BA 10 ADT !  
! * BA 20 MES '*****' !  
! * BA 30 MES 'DEBUT DE PROGRAMME : ' PROGR !  
! * BA 40 MES 'DATE D EXECUTION : ' DATOR !  
! * BA 50 MES 'BIBLIOTHEQUE : ' APPLI !  
! * BA 60 MES 'NUMERO DE SESSION : ' NUGNA !  
! * BA 70 MES 'GENERE LE : ' DATGN !  
! * BA 80 MES 'GENERE PAR : ' CODUTI !  
! * BA 90 MES '-----' !  
! * 90 N ARRET DU PROGRAMME                          10BL !  
! * 90 30 MES 'ANOMALIE SUR FICHER VSAM' !  
! * 90 40 MES 'PROBLEME A L OUVERTURE' 99IT 7-DP00-FONCT = 'I' !  
! * 90 45 MES 'PROBLEME SUR ENTREE SORTIE' 99IT 7-DP00-FONCT = 'L' !  
! * 90 50 MES '0C7 PROVOQUE' 99BL !  
! * 90 90 M SPACE 7-DP00-DUMP !  
! * 90 100 A 1 7-DP00-DUMP9 !  
!  
! *** FIN *** !  
! O: C1 CH: -P !  
! !  
-----
```

```
-----  
!  
!          DOCUMENTATION                                P0*JFL.LURE.TES.972 !  
! TRAITEMENTS PROGRAMME 1 P 2 TES001 test 001          FONCTION: 3 OA !  
!                                     4                                     5 !  
! A SS NLG OPE OPERANDE                                NVTY CONDITION !  
! 6 7 8 9 10                                         1112 13 !  
! . AC N DISPLAY STATUS FICHER AC                    10BL !  
! * BA N DISPLAY DU DEBUT                            10BL !  
! * BA 10 ADT !  
! * BA 20 MES '*****' !  
! * BA 30 MES 'DEBUT DE PROGRAMME : ' PROGR !  
! * BA 40 MES 'DATE D EXECUTION : ' DATOR !  
! * BA 50 MES 'BIBLIOTHEQUE : ' APPLI !  
! * BA 60 MES 'NUMERO DE SESSION : ' NUGNA !  
! * BA 70 MES 'GENERE LE : ' DATGN !  
! * BA 80 MES 'GENERE PAR : ' CODUTI !  
! * BA 90 MES '-----' !  
! . FB N DISPLAY STATUS FICHER FB                    10BL !  
! . XS N DISPLAY STATUS FICHER XS                    10BL !  
! * 90 N ARRET DU PROGRAMME                          10BL !  
!  
!  
!  
! CET ECRAN AFFICHE LES FONCTIONS GENEREES !  
! O: C1 CH: -PG !  
! !  
-----
```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| 1 | 1 | | <p>TYPE D'ENTITE</p> <p>Cette zone est saisissable, ce qui permet de copier des lignes d'un programme vers un écran et vice versa.</p> <p>O Ecran.</p> <p>P Programme.</p> |
| 2 | 6 | | CODE PROGRAMME OU ECRAN |
| 3 | 2 | | <p>CODE FONCTION (OBLIGATOIRE)</p> <p>Le code doit être choisi en fonction de l'emplacement désiré, en particulier lors de l'utilisation des modules Langage TP et Langage Batch dont les fonctions automatiques occupent des emplacements déterminés (Cf. Manuels de Référence).</p> <p>Dans une Macro-Structure, le code fonction peut être paramétré.</p> |
| 4 | 1 | | <p>CADRAGE DES OPERANDES</p> <p>Permet d'afficher à l'écran la partie droite des opérandes.</p> <p>BLANC Cadrage à gauche de la zone opérandes.</p> <p>AUTRE Cadrage à droite de la zone opérandes. Cette option interdit toute mise à jour.</p> |
| 5 | 1 | | <p>CADRAGE DE LA CONDITION</p> <p>Permet d'afficher à l'écran la partie droite de la condition.</p> <p>BLANC Cadrage à gauche de la zone condition.</p> <p>AUTRE Cadrage à droite de la zone condition. Cette option interdit toute mise à jour.</p> |
| 6 | 1 | | CODE ACTION |
| 7 | 2 | | <p>CODE SOUS-FONCTION</p> <p>Composé de chiffres ou lettres.</p> <p>Le code doit être choisi en fonction de l'emplacement désiré, comme le code fonction.</p> <p>Dans une Macro-Structure, le code sous-fonction peut être paramétré par un paramètre numérique.</p> |
| 8 | 3 | | <p>NUMERO DE LIGNE</p> <p>ZONE NUMERIQUE PARAMETRABLE</p> <p>Il est recommandé de numéroter les lignes de 10 en 10 pour permettre d'en insérer de nouvelles par la suite.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | \$n | Dans une Macro-Structure, il est possible de paramétrer les deux premiers caractères du numéro de ligne. |
| | | | <p>TRAITEMENT</p> <p>L'écriture des traitements spécifiques est basée sur l'utilisation d'un opérateur, suivi des opérandes correspondants.</p> <p>L'écriture des opérandes peut se poursuivre sur plusieurs lignes. Dans ce cas, l'opérateur ne doit pas être précisé sur les lignes suite.</p> <p>De manière générale, la ponctuation est gérée par PAC-BASE, essentiellement en fonction de la hiérarchisation des traitements. Cependant l'utilisateur a la possibilité de prendre en charge cette ponctuation.</p> <p>Dans une Macro-Structure, l'opérateur peut être paramétré.</p> |
| 9 | 3 | | <p>OPERATEURS DE TRAITEMENTS COURANTS</p> <p>N</p> <p>Titre de fonction ou sous-fonction. A mettre en tête de celle-ci, sur une seule ligne. La présence du titre permet de prendre les options par défaut pour le niveau hiérarchique.</p> <p>*</p> <p>Commentaire. Cet opérateur ne porte que sur la ligne où il figure. Le fait d'insérer des commentaires dans les traitements empêche la génération du point lorsque le commentaire est suivi d'un niveau 99. Pour faire générer ce point, insérer un '99BL' sur la première ligne de commentaires.</p> <p>M</p> <p>Transfert du premier opérande dans le(s) suivant(s).</p> <p>MA</p> <p>Transfert du premier opérande dans le(s) suivant(s) en utilisant l'ordre COBOL 'MOVE ALL'.</p> <p>P</p> <p>Branchement à l'étiquette de la fonction ou sous-fonction (Fffss) indiquée en premier opérande et retour en séquence après exécution de l'exit de la fonction ou sous-fonction (option par défaut), ou après exécution du paragraphe indiqué en deuxième opérande.</p> <p>C</p> <p>Calcul. Dans les opérandes la zone résultat doit être séparée des suivantes par le signe '='. Les zones intervenant dans le calcul doivent être séparées par les signes arithmétiques et les parenthèses nécessaires.</p> <p>A</p> <p>Addition du premier opérande dans le(s) suivant(s).</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | S | Le premier opérande est soustrait du second. |
| | | MP | Multiplication du premier opérande par le deuxième. |
| | | DV | Division du deuxième opérande par le premier. |
| | | MES | Emission des constantes ou paramètres définis en opérandes (DISPLAY). |
| | | ACC | Acceptation et transfert de paramètres ou constantes dans la zone définie en opérande (ACCEPT). |
| | | CAL | Appel du programme ou sous-programme défini dans la zone opérandes. (La clause USING, si elle est nécessaire, doit être écrite explicitement). |
| | | GT | Débranchement à la fin de la (sous-)fonction à laquelle appartient l'instruction et dont le niveau est indiqué sur deux chiffres en opérande. Si la (sous-)fonction est de type 'IT' et si elle utilisée avec une (sous-)fonction de type 'EL', l'opérateur GT à ce niveau renvoie au début de la (sous-)fonction de type 'EL'. |
| | | GFT | Abandon de l'itération de plus haut niveau en cours dans les programmes dont la structure est obtenue par des fonctions automatiques : Langage TP : débranchement à la fin du traitement des catégories, Langage Batch : débranchement à la fonction de fin de programme (F20) et positionnement des indicateurs de fin de traitement des fichiers. |
| | | GDI | Retour au début de l'itération de plus haut niveau en cours dans les programmes dont la structure est obtenue par des fonctions automatiques : DIALOGUE : passage à l'occurrence suivante de la catégorie en cours ou à la catégorie suivante, LANGAGE BATCH : retour au début de l'itération (F05). |
| | | GB | Retour en tête de la boucle. Permet d'abandonner l'itération en cours de niveau indiqué en opérande avec retour au début de la boucle (débranchement à l'étiquette Fffss-900). Si la (sous-)fonction est de type 'IT' et si elle utilisée avec une (sous-)fonction de type 'EL', l'opérateur GB à ce niveau renvoie à la fin de la (sous-)fonction de type 'EL'. Pour tous les opérateurs de débranchement, l'instruction générée se termine systématiquement par un point. |

| NUM | LON | CLASSE VALEUR EXA | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|-------------------------|---|
| | | EXA | Génère l'ordre COBOL 'EXAMINE' de la zone placée en opérande, éventuellement suivie de clauses complémentaires (TALLYING...); voir la syntaxe COBOL. |
| | | INS | Génère l'ordre COBOL 'INSPECT' de la zone placée en opérande, éventuellement suivie de clauses complémentaires (TALLYING...); voir la syntaxe COBOL. |
| | | COB | COBOL pur cadré en marge B du programme généré. |
| | | COA | COBOL pur cadré en marge A du programme généré. |
| | | U07 | COBOL pur cadré en colonne 7 du programme généré. |
| | | SUP | Empêche la génération de la (sous-)fonction automatique de même code que la ligne portant cet opérateur. Remarque : Pour le Module Batch, coder la valeur SUP sur la première ligne de la sous-fonction pour la supprimer. NB : Pour de plus amples informations concernant les opérateurs spécifiques, consulter les Manuels de Référence DIALOGUE GENERAL et LANGAGE BATCH. |
| | | SCH | Recherche, dans la table indiquée en premier opérande, de la Rubrique indiquée en deuxième opérande. Le deuxième opérande doit être sur une seule ligne. Le code de la table doit comporter le radical code working, s'il existe, (ex: 1-) et peut être suivi du code de la rubrique s'il est différent de celui de la Rubrique à rechercher. La Rubrique à rechercher doit obligatoirement être codifiée selon la normalisation des Rubriques PACBASE. Cet opérateur doit être employé dans une structure élémentaire explicite ou implicite de type bloc. La recherche se fera avec les indices standard associés à la table, et en cas d'insuccès, l'indice R sera supérieur à l'indice L. |
| | | SCB | Recherche dans une table triée. A la différence de l'opérateur 'SCH', la recherche est interrompue dès que l'argument de la table est supérieur à l'argument recherché. |
| | | | OPERATEURS DE TRAITEMENTS COBOLII |
| | | CON | Génère l'ordre COBOL 'CONTINUE' (pas d'opérande) |
| | | EVA | Génère l'ordre COBOL 'EVALUATE' de la condition exprimée par l'opérande. |
| | | EVT | Génère l'ordre COBOL 'EVALUATE TRUE' (en général pas d'opérande) |

| NUM | LON | CLASSE VALEUR EVF | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|-------------------------|---|
| | | | Génère l'ordre COBOL 'EVALUATE FALSE' (en général pas d'opérande) |
| | | EEV | Génère l'ordre COBOL 'END-EVALUATE' (pas d'opérande) |
| | | EIF | Génère l'ordre COBOL 'END-IF' (pas d'opérande) |
| | | EPE | Génère l'ordre COBOL 'END-PERFORM' (pas d'opérande) |
| | | ESE | Génère l'ordre COBOL 'END-SEARCH' (pas d'opérande) |
| | | INI | Génère l'ordre COBOL 'INITIALIZE' de la zone placée en opérande |
| | | SEA | Génère l'ordre COBOL 'SEARCH' de la condition exprimée par l'opérande |
| | | GOB | Génère l'ordre COBOL 'GO BACK' de la condition exprimée par l'opérande |
| | | STR | Génère l'ordre COBOL 'STRING' devant les paramètres indiqués dans la zone Opérande |
| | | UNS | Génère l'ordre COBOL 'UNSTRING' devant les paramètres indiqués dans la zone Opérande |
| | | | OPERATEURS DE TRAITEMENT DIALOGUE |
| | | | DEBRANCHEMENT |
| | | GF | Débranchement à la fin de la sous-fonction automatique où la ligne est insérée. Utilisable uniquement en fonctions 20, 25, 35 et 60. |
| | | GFR | Abandon des traitements de réception (débranchement à l'étiquette END-OF-RECEPTION). |
| | | GFA | Abandon des traitements d'affichage (débranchement à l'étiquette END-OF-DISPLAY). |
| | | GDB | Retour au début de l'itération en cours. Pour tous les opérateurs de débranchement, l'instruction générée se termine systématiquement par un point. (Il convient donc de veiller à ne pas renseigner 'EL' dans la zone 'Type de structure ou de condition' après une ligne utilisant un opérateur de débranchement.) |
| | | | APPEL ----- |
| | | OTP | Appel immédiat d'un nom externe. |
| | | OSC | Appel de l'écran indiqué en opérande. |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | OSD | Appel de l'écran indiqué en opérande après exécution des traitements de réception. LECTURE ET ECRITURE ----- Le code du Segment est indiqué en opérande. |
| | | XR | Lecture du Segment. |
| | | XP | Lecture du premier enregistrement en accès dynamique. Quel que soit le système, cet opérateur fournit toujours un enregistrement. |
| | | XRN | Lecture séquentielle du Segment (accès dynamique). |
| | | XRU | Lecture pour mise à jour du Segment. |
| | | XW | Ecriture du Segment. |
| | | XRW | Réécriture du Segment. |
| | | XD | Suppression du Segment. |
| | | XUN | Libération du Segment. L'utilisation de ces opérateurs peut entraîner la génération de la fonction d'accès correspondante. Dans le cas où le segment indiqué est une table ou une vue SQL, il faut s'assurer que le segment est défini dans l'écran avec une utilisation soit en réception soit en affichage. Les opérateurs XP et XRN sont réservés aux segments définis en catégorie répétitive avec une utilisation en affichage. DIVERS ----- |
| | | XAA | Suivi d'un code Segment (par exemple MV00), le Système génère automatiquement un perform de F80-MV00-AA. |
| | | YAA | Générateur d'une étiquette dans la fonction 80. Suivi d'un code Segment (par exemple MV00), le Système génère automatiquement une étiquette F80-MV00-AA. |
| | | ERU | Positionnement d'erreur utilisateur. La zone opérandes doit contenir : . code de l'erreur géré par l'utilisateur (4 car.), . un blanc, . éventuellement le code de la Rubrique sur six positions. |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | ERR | <p>Le libellé d'erreur correspondant au code indiqué sera affiché, le curseur sera placé sur la Rubrique mentionnée et elle prendra les attributs définis pour les Rubriques en erreur.</p> <p>Cet opérateur ne peut pas être utilisé sur une rubrique répétée.</p> <p>Positionnement d'erreur standard sur Rubrique.</p> <p>La zone opérandes doit contenir :</p> <ul style="list-style-type: none"> . code de l'erreur (1 car.). Peut être un caractère alphanumérique (sauf 0 et 1 réservés pour la codification des libellés documentaires), . un blanc, . code de la Rubrique variable à laquelle ce code erreur doit être associé. Le curseur est positionné et la Rubrique prend les attributs définis pour les Rubriques erronées. <p>Pour une Rubrique répétée, on indique l'indice en clair sur lequel porte l'erreur. Exemple : CORUB (2)</p> <p>L'utilisation des libellés d'erreur avec le module LANGAGE STRUCTURE est détaillée dans le manuel de référence correspondant.</p> |
| | | | <p>OPERATEURS DE TRAITEMENT BATCH</p> <p>Ces opérateurs ne sont pris en compte que dans la génération de programmes batch.</p> <p>Ils assurent les ordres d'entrée-sortie des fichiers en organisation séquentielle, séquentielle indexée ou dynamique.</p> <p>Les instructions générées sont adaptées à la variante du matériel et aux spécifications portées sur la ligne d'appel de la Structure de Données, à condition que l'appel soit effectué sur l'une des 23 premières lignes (on ne tient pas compte des lignes SUITE). Ainsi par exemple le même opérateur de lecture peut générer un READ AT END, un READ INVALID KEY, un CALL GETSEQ ou GETRAN, ou encore un RETURN AT END.</p> <p>A l'exception de l'ouverture et de la fermeture du fichier, ces opérateurs font intervenir la variable 'IK' qui aura une valeur différente de zéro en cas d'exécution anormale de l'instruction générée (fin de fichier, erreur sur clé, etc.). L'utilisateur aura la charge de déterminer l'action à entreprendre en fonction de la valeur de cette variable.</p> <p>L'opérande doit être sur la même ligne que l'opérateur car s'il est sur une ligne suite (opérateur absent), il sera généré dans la clause 'INVALID KEY'.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | OPE | Ouverture. |
| | | CLO | Fermeture. |
| | | R | Lecture. |
| | | W | Ecriture. |
| | | RW | Réécriture. |
| | | TRI | Tri. |
| | | RN | Read Next. |
| | | STA | Start. |
| | | DEL | Delete. |
| | | | <p>Pour les opérateurs 'OPE', 'CLO', 'R ', 'RN ' et 'DEL' le code S.D.-programme (2c) est le seul opérande.</p> <p>Pour l'écriture et la réécriture, la zone opérandes doit contenir le code enregistrement (FFEE) suivi éventuellement d'options telles que FROM ou AFTER.</p> <p>Pour le START, on indique le code S.D-programme, suivi éventuellement du positionnement de la clé au départ :</p> <p>Exemple : STA FF NOT < FF00-START génère</p> <pre> MOVE 0 TO IK START FF-FICHER KEY IS NOT < FF00-START INVALID KEY MOVE 1 TO IK. </pre> <p>Pour le tri les opérandes sont les paramètres suivant le verbe SORT.</p> |
| | | E | <p>Repère un libellé d'erreur utilisateur (Langage batch)</p> <p>La zone opérandes doit contenir :</p> <ul style="list-style-type: none"> . le type d'erreur sur 1 position (valeurs 0 à 5 interdites), . le code erreur sur 3 positions, . le code gravité d'erreur sur 1 position, . la première partie du libellé sur 27 positions. <p>La deuxième partie du libellé doit être codifiée dans la zone condition (32 positions).</p> |
| | | | OPERATEURS DE TRAITEMENT DE DATE |
| | | | TRAITEMENT DE DATE |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE ----- |
|-----|-----|------------------|--|
| | | ADT | <p>Appel de la date machine.</p> <p>La date sera fournie, quel que soit le matériel, sous la forme an, mois, jour dans la constante PACBASE 'DATOR', et éventuellement dans la zone écrite en opérande. Pour le matériel IBM, l'option inversion jour-mois dans la date, sur la fiche de la Bibliothèque, précise la forme sous laquelle se présente la date machine.</p> <p>Remarque : en COBOL II et COBOL 85, si l'année est inférieure à '61', le siècle est automatiquement initialisé à '20'.</p> |
| | | ADC | <p>Fournit la date machine sous la forme CCAAMMJJ où 'CC' représente le siècle.</p> <p>Remarque : en COBOL II et COBOL 85, si l'année est inférieure à '61', le siècle est automatiquement initialisé à '20'.</p> <p>Formatage de la date :</p> |
| | | AD | <p>Si la condition des opérateurs de traitement de date nécessite plusieurs lignes, les opérandes doivent être indiqués avant la suite de la condition (sans opérateur ou sur des lignes commentaire).</p> <p>Une date peut être formatée de différentes façons :</p> <p>Les opérandes utilisables sont 'XY CORUB1 CORUB2' où 'X' et 'Y' représentent les codes du tableau ci-dessous.</p> <p>-----</p> <p>! CODE! FORMAT GENERE !</p> <p>-----</p> <p>! I ! AN, MOIS, JOUR !</p> <p>-----</p> <p>! D ! JOUR, MOIS, AN ou MOIS, JOUR, AN selon le ! ! ! type de format demandé dans le Dictionnaire. !</p> <p>-----</p> <p>! E ! JOUR/MOIS/AN ou MOIS/JOUR/AN (Etendu) !</p> <p>-----</p> <p>! S ! SIECLE, AN, MOIS, JOUR !</p> <p>-----</p> <p>! C ! MOIS, JOUR, SCLE, AN ou JOUR, MOIS, SCLE, AN !</p> <p>-----</p> <p>! M ! MOIS/JOUR/SIECLE AN ou JOUR/MOIS/SIECLE AN !</p> <p>-----</p> <p>! G ! SIECLE AN - MOIS - JOUR !</p> <p>-----</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | <p>EXEMPLE :</p> <p>Pour transformer une date de format 'I' en date de format 'D', il convient d'indiquer 'AD' dans la zone opérateur et :</p> <p style="text-align: center;">'ID CORUB1 CORUB2'</p> <p>dans la zone opérande, CORUB1 étant la Rubrique contenant une date en an, mois, jour (il est possible d'utiliser la constante PACBASE 'DATOR' si besoin), et CORUB2 étant la Rubrique date transformée en jour, mois, an.</p> <p>Pour le module batch, la fonction de transformation des dates est générée en F9520. Vous pouvez la modifier en codant, sur une ligne de type 'O' des -G du Programme, le paramètre DATPRO = ffss, où ffss est le code fonction-sous fonction choisi.</p> |
| | | ADO | Alimentation du siècle à partir d'une zone DAT-CTY initialisée à '19' et modifiable. |
| | | AD1 | Alimentation du siècle à '19' si l'année de la date considérée est inférieure à la fenêtre DAT-CTYT (61 par défaut), à '20' dans le cas contraire. |
| | | AD2 | Alimentation du siècle à '20' si l'année de la date considérée est inférieure à la fenêtre DAT-CTYT ('61' par défaut) ou à '19' dans le cas contraire. |
| | | | <p>TRAITEMENT DE DATE (Dialogue)</p> <p>-----</p> |
| | | AD6 | <p>ADT + ADI</p> <p>Fournit la date machine sous la forme JJMMAA ou MMJJAA en fonction du FORMAT DE LA DATE indiqué dans la définition de la bibliothèque dans la zone des 6 caractères indiquée en opérande.</p> |
| | | AD8 | <p>ADT + ADI + ADE</p> <p>Fournit la date machine sous la forme JJ/MM/AA ou MM/JJ/AA en fonction du FORMAT DE LA DATE indiqué dans la définition de la Bibliothèque dans la zone de 8 caractères indiquée en opérande.</p> |
| | | | <p>TRAITEMENT DE DATE (BATCH)</p> <p>-----</p> |
| | | ADI | <p>Inversion d'une date (tous modules sauf Dialogue): les deux premiers et les deux derniers caractères sont inversés.</p> <p>Les deux opérandes sont supposés avoir une longueur de</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | <p>six caractères. Le deuxième opérande n'est pas obligatoire, dans ce cas, le premier opérande récupère la date inversée.</p> |
| | | ADS | <p>Fournit la date inversée avec le siècle (tous modules sauf Dialogue). Les deux opérandes sont supposés avoir une longueur de huit caractères. Le deuxième opérande n'est pas obligatoire, dans ce cas le premier opérande récupère la date inversée.</p> |
| | | ADE | <p>Mise en forme d'une date par insertion de slash ('/') (tous modules sauf Dialogue). Le premier opérande contient la date sur six caractères, le deuxième recevra la date mise en forme et doit avoir une longueur de huit caractères.</p> |
| | | ADM | <p>Mise en forme d'une date avec siècle par insertion de slash ('/') ; (tous modules sauf Dialogue).</p> |
| | | | <p>TRAITEMENT DE L'HEURE -----</p> |
| | | TIM | <p>Fournit l'heure sous la forme HHMMSS à partir de la zone EIBTIME pour CICS, et TIME pour les autres matériels. Exemple : A SS NLG OPE OPERANDE NVTY CONDITION BB 100 TIM ZONE1</p> |
| | | TIF | <p>Transforme le format de l'heure HHMMSS en HH:MM:SS. Exemple : A SS NLG OPE OPERANDE NVTY CONDITION BB 100 TIF ZONE1 ZONE2</p> |
| | | | <p>OPERATEURS DE COMMUNICATION</p> <p>NOTE: Non opérationels avec le Module Dialogue.</p> <p>Ils assurent les liaisons entre un programme COBOL et les 'UNITES DE COMMUNICATION'.</p> |
| | | ENA | <p>ENABLE connexion de l'unité.</p> |
| | | DSB | <p>DISABLE déconnexion de l'unité.</p> |
| | | RE | <p>RECEIVE réception d'un message.</p> |
| | | SD | <p>SEND émission d'un message.</p> |
| | | | <p>OPERATEURS DES S.G.B.D.</p> <p>Opérateurs IDS 1. -----</p> <p>Non opérationnel avec le module Dialogue.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | Ils se caractérisent par une première lettre 'T' ou 'T' suivie de deux caractères, définis ci-dessous, spécifiques de l'instruction. |
| | | MV | MOVE |
| | | MC | MOVE CURRENT |
| | | MD | MODIFY |
| | | DE | DELETE |
| | | R | RETRIEVE |
| | | SE | SET |
| | | ST | STORE |
| | | I | IF ----- GO TO --- (NOM-PARAGRAPH) |
| | | IE | IF ERROR GO TO --- (NOM-PARAGRAPH) |
| | | HE | HEAD |
| | | CL | CLOSE |
| | | OU | OPEN UPDATE |
| | | OR | OPEN RETRIEVAL |
| | | | Les paramètres ou options de ces instructions sont à indiquer dans la zone opérandes. |
| | | | 'ENTER IDS' sera généré pour tout opérateur commençant par la lettre 'T'; un point sera généré à la fin d'une série d'opérateurs IDS ou devant un opérateur commençant par 'T'. |
| | | | Certains de ces opérateurs peuvent être utilisés en TDS (nature du programme 'T'), dans ce cas 'ENTER TDS' sera généré à la place de 'ENTER IDS'. |
| | | | Opérateurs CODASYL ----- |
| | | | Ils se caractérisent par une première lettre 'B' suivie de deux caractères, définis ci-dessous, spécifiques de l'instruction. |
| | | RY | READY |
| | | FH | FINISH |

| NUM | LON | CLASSE VALEUR FD | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE FIND |
|-----|-----|------------------------|--|
| | | G | GET |
| | | ER | ERASE |
| | | DT | DISCONNECT --- FROM --- |
| | | CT | CONNECT --- TO --- |
| | | MD | MODIFY |
| | | ST | STORE |
| | | | Opérateur SOPHOCLE. ----- |
| | | | Non opérationnel avec le module Dialogue. |
| | | SOP | < ---- >; Une ligne commentaire terminée par un ';' sera générée au début de toute série d'opérateurs SOPHOCLE. |
| | | | OPERATEURS POUR MONITEURS T.P. Ces opérateurs ne sont pas utilisables pour les descriptions de traitements d'Ecran avec le module DIALOGUE. Opérateurs CICS MACRO LEVEL. ----- Ils se caractérisent par une première lettre, 'D', suivie de deux caractères spécifiques de la macro CICS |
| | | DXX | DFHXX TYPE= |
| | | DBM | DFHBMS TYPE= |
| | | | Tous les opérandes doivent être cadrés à gauche de la zone. Le caractère de continuation en colonne 72 du programme généré et le cadrage des lignes suite sont gérés automatiquement par PACBASE. |
| | | | Opérateurs CICS COMMAND LEVEL ----- |
| | | EXC | EXEC CICS opérandes END-EXEC. Opérateurs TDS sur H66 BCD. ----- |
| | | | Non opérationnel avec le module Dialogue. |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>Ils se caractérisent par une première lettre 'T' ou 'T' suivie de deux caractères, définis ci-dessous, spécifiques de l'instruction.</p> <p>MV MOVE</p> <p>MC MOVE CURRENT</p> <p>DE DELETE</p> <p>MD MODIFY</p> <p>R RETRIEVE</p> <p>SE SET</p> <p>ST STORE</p> <p>I IF ----- GO TO --- (NOM-PARAGRAPH)</p> <p>IE IF ERROR GO TO --- (NOM-PARAGRAPH)</p> <p>SD SEND</p> <p>SB SENDB</p> <p>RE RECEIVE</p> <p>T TERMINATE-TPR</p> <p>TL TERMINATE-TPR AND LOAD</p> <p>Les paramètres ou options de ces instructions sont à indiquer dans la zone opérandes.</p> <p>'ENTER TDS' sera généré pour tout opérateur commençant par la lettre 'T'; un point sera généré à la fin d'une série d'opérateurs TDS ou devant un opérateur commençant par 'T'.</p> <p>Certains de ces opérateurs peuvent être utilisés en IDS (nature du programme différente de 'T'), dans ce cas 'ENTER IDS' sera généré à la place de 'ENTER TDS'.</p> |
| | | EXP | <p>OPERATEUR P.A.F.</p> <p>Cet opérateur est utilisé pour activer une requête d'accès à la base VA Pac à l'aide du module P.A.F. (PACBASE Access Facility). Il génère des modules PAF ou DAF.</p> <p>EXEC PAF (requête..) END-EXEC.</p> |

| NUM LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE Pour plus de détails, voir le Manuel de Référence "P.A.F." ou "DSMS Access Facility". |
|---------|---------------|--|
| | | <p>OPERATEURS SQL</p> <p>Les opérateurs SQL sont documentés en détail dans le Manuel de Référence "Description de Bases de Données Relationnelles", Chapitre "Accès SQL".</p> <p>SCC Ordre CONNECT ou son équivalent.</p> <p>SDC Ordre DISCONNECT ou son équivalent.</p> <p>SCO Ordre COMMIT.</p> <p>SRO Ordre ROLLBACK.</p> <p>SWH Ordre WHENEVER.</p> <p>La syntaxe doit être la suivante :</p> <pre> SCC ccccc d SDC ccccc d r SCO ccccc d SRO ccccc d SWH instruction </pre> <p>cccccc : code Pacbase du bloc (sur 6 caractères). d : valeur 2 si base distribuée (ex : Oracle sybase). r : valeur R pour choisir le DISCONNECT avec ROLLBACK.</p> <p>Règles :</p> <ul style="list-style-type: none"> - Les indicateurs d et r peuvent être inversés. - Chaque ordre peut être complété par les lignes suivantes (sans opérateur). Vous pouvez indiquer, par exemple, l'option FORCE dans un ordre COMMIT dans ORACLE. <p>Génération :</p> <ul style="list-style-type: none"> - Si dans les Appels de Segments (-CS pour le Module Dialogue ou -CD pour le Module Batch), vous déclarez une organisation SQL (zone ORGANISATION) et donc un code Bloc (zone NOM EXTERNE), cette organisation est prioritaire par rapport au type de bloc indiqué sur la Définition du Bloc. - Si le bloc figure dans les Appels de Segments comme étant distribué, les ordres liés à ce bloc seront générés "distribué". - Les ordres inconnus d'un SQL sont ignorés. <p>La génération du END-EXEC est automatique et, pour le générateur batch, le END-EXEC est toujours suivi d'un point.</p> |

| NUM LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|---------|---------------|--|
| | SQL | <p>Pour batch uniquement, l'ordre SQL obtenu est standard avec les possibilités de personnalisation introduites dans le -G du segment.</p> <p>La personnalisation des accès SQL est documentée dans le Manuel "Description de bases de données relationnelles", Chapitre "Accès SQL", Sous-chapitre "Personnalisation des accès SQL".</p> <p>Pour prendre en compte les accès personnalisés, le -CD du Programme doit contenir le code du Bloc dans la zone NOM EXTERNE et l'organisation dans ORGANISATION.</p> <p>L'opérateur SQL doit avoir la syntaxe suivante :</p> <p>OPE I OPERANDES SQL I FFSS FBSS OS OP (format 1) SQL I FFSS OS OP (format 2)</p> <p>FFSS : code fichier-segment dans le programme. FBSS : code fichier-segment bibliothèque. OS : type de l'ordre standard (sur deux caractères) ou ordre spécifique décrit dans le S FBSS G. OP : ordre particulier décrit dans les -G du Segment</p> <p>Le format 1 doit être utilisé quand le code programme est différent du code bibliothèque. Le code de l'ordre particulier est facultatif, il vient modifier la description obtenue par l'ordre standard.</p> <p>Les organisations SQL retenues sont :</p> <p>C (Interel) M (Datacom) N (Nonstop) O et P (Oracle) Q (SQL-DS ALLbase) 2 (DB2) 3 (SQL SERVER) 4 (DB2/400) 9 (Informix Ingres et SYBASE) H (compatibilité code Pacbench C/S)</p> <p>Pour certaines organisations, il faut avoir la référence au type de bloc (H et 9 par exemple).</p> <p>Restrictions :</p> <ul style="list-style-type: none"> - la syntaxe pour les ordres des blocs RDMS n'est pas implémentée (organisation R). - la règle de préfixage n'est pas appliquée. On conserve le nom de la table tel qu'il est avec suppression du point de tête s'il y en a un. <p>Limite :</p> <ul style="list-style-type: none"> - dans les recouvrements programme-macro et |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>macro-macro, les lignes générées obtenues par le S FFSS G ne peuvent pas être supprimées.</p> <p>Opérateur relationnel ----- EXEC SQL opérandes END-EXEC.</p> |
| 10 | 32 | EXQ | <p>OPERANDES</p> <p>Le paramétrage de cette zone dépend de l'opérateur qui lui est associé.</p> <p>Le premier opérande doit être obligatoirement sur la ligne portant l'opérateur. Lorsqu'une opération fait intervenir divers facteurs, ceux-ci doivent être indiqués à la suite l'un de l'autre séparés par au moins un blanc.</p> <p>La qualification des noms-données par 'OF' est acceptée à la condition que le 'OF' du premier opérande soit sur la ligne portant l'opérateur.</p> <p>Dans le cas où le premier opérande est un littéral alphanumérique qui ne peut tenir intégralement sur une ligne, la suite du littéral peut être indiquée entre quotes sur la ligne suivante, PACBASE assurera la continuité au niveau de la génération.</p> |
| | | | <p>STRUCTURE</p> <p>Toute structure (fonction ou sous-fonction) a un nombre niveau et un type, auxquels on associe éventuellement une condition, une variable ou une valeur.</p> <p>Les structures de niveau différent de 99 doivent être obligatoirement définies sur la première ligne d'une (sous)-fonction, le conditionnement pouvant se poursuivre sur plusieurs lignes.</p> <p>Les valeurs par défaut concernant le niveau et le type sont prises sur la ligne titre de la (sous-) fonction.</p> <p>Les fonctions constituent les structures de plus haut niveau (05).</p> |
| 11 | 2 | | <p>NIVEAU HIERARCHIQUE</p> <p>ZONE NUMERIQUE PARAMETRABLE</p> <p>S'indique uniquement sur la première ligne d'une structure.</p> <p>La condition d'exécution d'une structure de niveau donné s'applique à toutes les structures de niveaux inférieurs apparaissant après elle dans la séquence du programme jusqu'à ce que l'on rencontre une structure de niveau égal ou supérieur.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | 05 | Niveau fixe attribué aux fonctions, pris par défaut sur la ligne titre à condition qu'il en existe une. |
| | | 10 | Niveau pris par défaut pour une sous-fonction sur la ligne titre à condition qu'il en existe une. |
| | | 06 A 98 | Niveaux possibles pour une sous-fonction. |
| | | 99 | Définit une structure élémentaire à l'intérieur d'une fonction ou d'une sous-fonction. Dans une Macro-Structure, le niveau hiérarchique peut être paramétré. REMARQUE : L'insertion de commentaires sur des lignes de traitement de niveau '99' et de type 'IT' empêche la génération du point de fin de phrase. Il est par contre possible d'introduire du commentaire sur des lignes de niveau '99' et de type 'BL'. |
| 12 | 2 | | <p>TYPE DE STRUCTURE OU DE CONDITION</p> <p>Sur la première ligne de définition d'un bloc de traitement, indique le type du bloc conformément aux règles de la programmation structurée.</p> <p>Une (sous-) fonction constitue un bloc de traitement ou structure. Mais on peut également, à l'intérieur d'une (sous-) fonction définir des structures élémentaires caractérisées par un niveau 99.</p> <p>Sur les autres lignes définit une suite de condition.</p> <p>BL Structure de type 'BLOCK'. Valeur par défaut pour toute structure non conditionnée.</p> <p>IT Structure de type 'IF THEN'. Exécutée si son conditionnement est satisfait. Valeur par défaut pour toute structure conditionnée.</p> <p>EL Structure de type 'ELSE'. Interdite au niveau fonction. Exécutée si la structure précédente de même niveau (et nécessairement de type 'IF THEN') ne l'a pas été. Une telle structure ne peut avoir de condition propre.</p> <p>CO Structure de type 'CASE OF'. Interdite au niveau fonction et au niveau 99. Permet de réaliser l'aiguillage multibranches, c'est à dire des traitements exclusifs les uns des autres,</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>exécutés suivant les valeurs possibles d'une variable. Sur une structure de type 'CASE OF' on définit uniquement le nom de la variable (dans la partie condition et sur une seule ligne), les valeurs possibles de cette variable sont précisées sur les structures (non élémentaires) de niveau hiérarchique immédiatement inférieur. Dans un écran Dialogue, l'imbrication de structure de type 'CASE OF' n'est pas autorisée.</p> |
| | | DW | <p>Structure de type 'DO WHILE'. Interdite au niveau fonction et au niveau 99. Structure exécutée de façon répétitive tant que son conditionnement est satisfait.</p> |
| | | DU | <p>Structure de type 'DO UNTIL'. Interdite au niveau fonction et au niveau 99. Structure exécutée de façon répétitive jusqu'à ce que son conditionnement soit satisfait. Elle est donc exécutée au moins une fois. Pour ces deux derniers types de structure il convient de veiller à faire évoluer les paramètres du conditionnement pour éviter de boucler.</p> |
| | | DO | <p>Structure de type 'DO' (structure de type boucle). Interdite au niveau fonction et 99. Structure exécutée de façon répétitive entre 2 bornes (précisées dans la zone 'Conditionnement') en fonction d'un pas positif ou négatif. Le pas est facultatif. Les bornes peuvent être soit une valeur, soit un nombre (signé ou non). Elles doivent figurer sur la ligne titre de la sous-fonction, séparées par un espace. Les paramètres doivent être renseignés dans l'ordre suivant: Borne inférieure (positive), borne supérieure et le pas d'incrément de l'indice. DO exige 3 paramètres, dont les 2 premiers sont obligatoires. Tous les 3 doivent paraître sur la première ligne. PACBASE génère automatiquement un indice JffnnR, 'ffnn' représentant le code (sous-)fonction.</p> |
| | | OR | <p>Suite de conditionnement associée aux lignes précédentes par un 'OU' logique.</p> |

| NUM | LON | CLASSE VALEUR AN | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------------|---|
| | | AN | <p>Suite de conditionnement associée aux lignes précédentes par un 'ET' logique.</p> <p>N-B: les parenthèses regroupant des termes dans une condition doivent être indiquées dans le texte de la condition.</p> <p>Dans une Macro-Structure le type de structure ou de condition ne peut pas être paramétré.</p> |
| | | WH | <p>Vous pouvez utiliser des ordres COBOL derrière un ordre EVA ou SEA.</p> <p>Chacun des ordres indique le traitement à effectuer pour une condition particulière.</p> <p>Dans l'exemple suivant, le traitement t1 est effectué lorsque la condition c1 est remplie.</p> <pre>EVA ----- ---t1--- 99WH ---c1--- ---t2--- 99WH ---c2--- ---t3--- 99WH ---c3---</pre> <p>DIALOGUE -----</p> <p>Les types suivants sont réservés aux modules DIALOGUE et PACBENCH C/S :</p> |
| | | *A | <p>Insertion de la sous-fonction avant une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.</p> |
| | | *P | <p>Insertion de la sous-fonction après une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite. En fonction 20 indiquer le code.</p> |
| | | *R | <p>Remplacement d'une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.</p> <p>Pour plus de détails concernant ces trois types, se reporter au Manuel de Référence Dialogue ou Pacbench C/S, Chapitre 'Emploi du langage structuré', Sous-chapitre 'Traitements spécifiques'.</p> <p>Les deux types suivants sont réservés au module Pacbench C/S (composants serveur uniquement) :</p> |
| | | *C | <p>Insertion ou remplacement des traitements sur serveur ou vue logique. Il ne peut être indiqué que sur un niveau 05.</p> |
| | | *B | <p>Insertion dans les traitements élémentaires appelés par PERFORM.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| | | | <p>Pour plus de détails concernant ces deux types, se reporter au Manuel de Référence Pacbench C/S, Chapitre 'Emploi du langage structuré', Sous-chapitre 'Traitements spécifiques'.</p> |
| 13 | 28 | | <p>CONDITION D'EXECUTION</p> <p>Exprimé sans 'IF', ni 'AND', ni 'OR', ni 'GO TO', ni point.</p> <p>La première ligne d'une condition doit obligatoirement être associée à un niveau hiérarchique explicite ou implicite (05 pour une fonction, 06 à 98 pour une sous-fonction, 99 pour une structure élémentaire à l'intérieur d'une fonction ou sous-fonction). Par contre sur les lignes suite, il ne doit pas être précisé de niveau hiérarchique.</p> <p>Pour une structure de type 'CASE OF' le nom de la variable sujet de l'alternative doit être indiqué sur une seule ligne.</p> <p>Pour les structures dépendant d'un 'CASE OF', la condition d'exécution représente une valeur possible de la variable sujet du 'CASE OF'.</p> <p>Dans une macro-structure, la condition d'exécution peut être paramétrée.</p> |

6. TITRES, CONDITIONS ET NIVEAUX DES TRAITEMENTS

6.1. TITRES ET CONDITIONS DES TRAITEMENTS (-TC)

AFFICHAGE DES TITRES ET CONDITIONNEMENTS (-TC)

L'écran "TITRES ET CONDITIONNEMENTS" constitue une aide à la programmation.

Le problème majeur qui se pose aux programmeurs qui débutent dans l'utilisation de VA Pac est de maîtriser ses possibilités de génération, c'est-à-dire de prévoir les traitements qui seront générés dans un Programme.

En effet, le travail de programmation lié à l'utilisation de VA Pac comporte deux étapes :

- . Une première étape de mise en oeuvre des fonctions automatiques du système,
- . Une deuxième étape de "personnalisation" du programme généré.

La première étape est assurée par l'accès aux données dans les Programmes batch ou conversationnels. Elle est mise en oeuvre par :

- . L'appel des Structures de Données pour les Programmes batch (choix : P.....CD),
- . L'appel de Segments dans les Programmes conversationnels ou Ecrans (choix : O.....CS),

Les informations contenues dans ces appels conditionnent la (non-)génération de certaines (sous-)fonctions automatiques des programmes générés.

Par exemple :

Pour un Programme batch, l'utilisation 'M' pour une S.D. entraîne la génération des fonctions de contrôle.

Pour un Programme conversationnel (écran), l'utilisation en réception 'E' pour un segment entraîne la génération des accès au Segment avec positionnement d'une erreur.

La deuxième étape est assurée par l'introduction de Langage Structuré. Elle permet de compléter les traitements automatiques générés par PACBASE en ajoutant certains éléments spécifiques du Programme. Elle est mise en oeuvre par :

- . L'appel de M.S.P. pour les traitements standardisés,

(Choix : P.....CP ou O.....CP)

. L'appel direct de Langage Structuré

(choix : P.....P ou O.....P).

L'ensemble "Traitements Automatiques + Langage Structuré" va constituer l'ensemble du Programme.

L'écran "Titres et conditionnements" permet de visualiser le titre et le conditionnement de tous les traitements d'un Programme batch ou conversationnel (écran), c'est à dire à la fois les traitements automatiquement générés et les traitements spécifiques.

Par l'intermédiaire de cet écran, l'utilisateur peut donc connaître la structure générale du Programme et donc déceler les erreurs liées à l'utilisation des différents éléments constitutifs du Programme.

MISE EN OEUVRE

Cet écran est obtenu par les choix :

'PaaaaaaTCffxx<nn' ou

'Paaaaaa<nnTCffxx' pour les programmes,

'OssssssTCffxx<nn' ou

'Oaaaaaa<nnTCffxx' pour les écrans.

Avec :

```
.'aaaaaa' = code programme  
.'ssssss' = code écran  
.'ff'     = code fonction (par défaut : ' ' )  
.'xx'     = code sous-fonction (par défaut : ' ' )  
.'nn'     = nombre niveau (par défaut : '05')
```

Exemple :

Pour obtenir la liste de tous les titres et conditionnements du programme 'commun', l'utilisateur doit entrer le choix :

Ppgm001TC

Si, pour éviter d'avoir une image trop détaillée du programme l'utilisateur veut l'affichage des titres et conditionnements jusqu'au niveau 15, à partir de la fonction 'ff', sous-fonction 'xx', il doit entrer le choix :

'Ppgm001TCffxx<15' ou 'Ppgm001<15TCffxx'.

REPERAGE DES LIGNES AFFICHEES

Les lignes affichées sur cet écran peuvent provenir de trois sources :

1-Génération automatique VA Pac :

Dans ce cas, les lignes affichées proviennent des lignes '-CD' pour les programmes ou '-CS' pour les écrans, elles-mêmes introduites par l'utilisateur. Elles sont repérées par un '.' dans le code action.

2-Appels de macro-structures :

Les lignes affichées proviennent des lignes '-CP' (programmes et écrans). Elles sont repérées par le caractère '*' dans le code action.

3-Langage Structuré introduit directement :

Ces lignes sont repérées par un blanc dans le code action.

POSSIBILITES DE MISE A JOUR

L'écran '-TC' affiche des titres de (sous-)fonctions avec leur conditionnement. Les mises à jour ne peuvent donc s'appliquer qu'à l'ensemble d'une (sous-)fonction.

Traitements automatiques et macro-structures :

La seule possibilité de mise à jour est la suppression de la génération des traitements concernés (fonction ou sous-fonction).

Cette suppression se fait grâce à l'opérateur 'S' qui correspond à l'opérateur 'SUP' des lignes '-P'.

Ces traitements sont repérés par un code action '.' ou '*' qui inhibent la mise à jour. Pour assurer la prise en compte des modifications, l'utilisateur doit donc effacer ce code action.

Traitements spécifiques :

L'utilisateur peut créer, modifier ou annuler des titres de (sous-)fonctions introduits grâce au langage structuré. Pour cela, il dispose des codes actions 'C', 'M', 'A' ou 'blanc'. Il peut modifier le titre, le niveau, le type et le conditionnement d'une (sous-)fonction.

DEBRANCHEMENT SUR UNE (SOUS-)FONCTION AFFICHEE

L'utilisateur a la possibilité de se débrancher sur l'écran '-P' correspondant à une ligne affichée. Pour cela, il doit placer le curseur sur la ligne concernée et utiliser la touche fonction réservée à cet effet (en standard PF10).

De même, il peut se débrancher sur l'écran '-PG' à partir de la ligne repérée par le curseur grâce à la touche fonction réservée à cet effet (en standard PF9).

Enfin, il peut demander l'affichage du même écran à partir de la ligne repérée par le curseur à l'aide de la touche fonction définie à cet effet (en standard PF8 sauf en IMS).

OPERATEURS ADMIS EN MISE A JOUR DANS CET ECRAN

| | |
|-------|---|
| N | Note (Titre). |
| S | Suppression (équivalent à 'SUP'). |
| Blanc | Suite du conditionnement d'une (sous-)fonction. |

Tout autre opérateur est refusé.

CONDITIONS PREREQUISES

Le programme ou l'écran doivent avoir été définis.

Remarque concernant le Module DIALOGUE.

Des différences entre le contenu de l'écran "-TC" et le programme généré peuvent apparaître. Elles tiennent aux raisons suivantes:

Fonction F80 :

Sur l'écran "-TC", l'ordre des sous-fonctions dépend du code des Segments dans le Programme alors que dans le généré il dépend du code des Segments dans la Bibliothèque.

Pour un Segment appelé sur l'écran "-CS", d'organisation différente de 'U' et non utilisé en affichage ou en réception, l'affichage est quand même simulé sur l'écran "-TC".

Fonction F81 :

Le contrôle de numéricité et le contrôle de date étant considérés comme toujours générés, les sous-fonctions F8110 et F8120 s'affichent toujours sur l'écran "-TC". Or, la sous-fonction F8110 n'est générée que si des rubriques de nature variable et de format numérique sont appelées. Quant à la sous-fonction F8120 elle n'est générée qu'en cas d'appel de rubriques de nature variable et de format date.

TITRES, CONDITIONS ET NIVEAUX DES TRAITEMENTS
 TITRES ET CONDITIONS DES TRAITEMENTS (-TC)

6
 1

```

-----
!
!           VALIDATION MATERIEL IBM                PO*AG.LILI.IBM.847 !
!
! TRAITEMENTS PROGRAMME 1 FO20MA MISE A JOUR FOURNISSEURS !
!
! A FOSS NLG O OPERANDE                        NVTY CONDITION !
! 2 3 4 5 6 7                                8 9 10 !
! . 01      N INITIALISATIONS                  05BL !
! . 05      N LECTURE FICH. SEQ. SS RUPTURE    05BL !
! * 05FB    S !
! . 10      N LECTURE FICH. SEQ. A RUPTURE    05BL !
! * 10FB    S !
! . 19      N VIDAGE DE LA TABLE              05IT FT = ALL '1' !
! . 20      N FIN DE TRAITEMENT                05IT FT = ALL '1' !
! . 22      N CALCUL DERNIERS ENREG.          05BL !
! * 24      S !
! * 26      10 N CALCUL RUPTURES TOTALES      05BL !
! . 30      N CALCUL VARIABLES CONTROLE      05IT MF-CF1 = '1' !
! . 33      N CONTROLE IDENTIFICATION         05IT MF-CF1 = '1' !
! . 36      N CONTROLE ENREGISTR. EN DOUBLE  05BL !
! . 39      N DETERMINATION PRESENCE RUBR.    05IT MF-CF1 = '1' AND EN-EI = '0' !
! . 42      N CONTROLE DE STRUCTURE ENREG.    05IT MF-CF1 = '1' AND EN-EI = '0' !
! . 45      N CONTROLE CONTENU RUBRIQUES     05IT MF-CF1 = '1' AND EN-EI = '0' !
! . 46      N AUTRES CONTROLES                05BL !
! . 51      N CONTROLE PRESENCE ENREG.       05IT MF-CF1 = '1' AND EN-EI = '0' !
! SIMULATION LIMITEE AU NIVEAU: 05 !
! O: C1 CH: -TC !
!
-----

```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| 1 | 6 | | CODE PROGRAMME OU ECRAN |
| 2 | 1 | | CODE ACTION |
| 3 | 2 | | CODE FONCTION Le code doit être choisi en fonction de l'emplacement désiré, en particulier lors de l'utilisation des modules Langage TP et Langage Batch dont les fonctions automatiques occupent des emplacements déterminés (Cf. Manuels de Référence). Dans une Macro-Structure, le code fonction peut être paramétré. |
| 4 | 2 | | CODE SOUS-FONCTION Composé de chiffres ou lettres. Le code doit être choisi en fonction de l'emplacement désiré, comme le code fonction. Dans une Macro-Structure, le code sous-fonction peut être paramétré par un paramètre numérique. |
| 5 | 3 | \$n | NUMERO DE LIGNE ZONE NUMERIQUE PARAMETRABLE Il est recommandé de numéroter les lignes de 10 en 10 pour permettre d'en insérer de nouvelles par la suite. Dans une Macro-Structure, il est possible de paramétrer les deux premiers caractères du numéro de ligne. |
| 6 | 1 | N S Blanc | TRAITEMENT Les seuls opérateurs utilisables sur l'écran -TC sont: N Titre de (sous-)fonction. S Equivalent de l'opérateur 'SUP': suppression d'une (sous-)fonction automatique. Blanc Suite de condition d'exécution. |
| 7 | 32 | | OPERANDES Titre de la (sous-)fonction. |
| 8 | 2 | | NIVEAU HIERARCHIQUE ZONE NUMERIQUE PARAMETRABLE S'indique uniquement sur la première ligne d'une structure. La condition d'exécution d'une structure de niveau donné s'applique à toutes les structures de niveaux inférieurs apparaissant après elle dans la séquence |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | <p>du programme jusqu'à ce que l'on rencontre une structure de niveau égal ou supérieur.</p> <p>05 Niveau fixe attribué aux fonctions, pris par défaut sur la ligne titre à condition qu'il en existe une.</p> <p>10 Niveau pris par défaut pour une sous-fonction sur la ligne titre à condition qu'il en existe une.</p> <p>06 A 98 Niveaux possibles pour une sous-fonction.</p> <p>99 Définit une structure élémentaire à l'intérieur d'une fonction ou d'une sous-fonction.</p> <p>Dans une Macro-Structure, le niveau hiérarchique peut être paramétré. REMARQUE : L'insertion de commentaires sur des lignes de traitement de niveau '99' et de type 'IT' empêche la génération du point de fin de phrase. Il est par contre possible d'introduire du commentaire sur des lignes de niveau '99' et de type 'BL'.</p> |
| 9 | 2 | | <p>TYPE DE STRUCTURE OU DE CONDITION</p> <p>Sur la première ligne de définition d'un bloc de traitement, indique le type du bloc conformément aux règles de la programmation structurée.</p> <p>Une (sous-) fonction constitue un bloc de traitement ou structure. Mais on peut également, à l'intérieur d'une (sous-) fonction définir des structures élémentaires caractérisées par un niveau 99.</p> <p>Sur les autres lignes définit une suite de condition.</p> <p>BL Structure de type 'BLOCK'. Valeur par défaut pour toute structure non conditionnée.</p> <p>IT Structure de type 'IF THEN'. Exécutée si son conditionnement est satisfait. Valeur par défaut pour toute structure conditionnée.</p> <p>EL Structure de type 'ELSE'. Interdite au niveau fonction. Exécutée si la structure précédente de même niveau (et nécessairement de type 'IF THEN') ne l'a pas été. Une telle structure ne peut avoir de condition propre.</p> <p>CO Structure de type 'CASE OF'. Interdite au niveau fonction et au niveau 99.</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | <p>Permet de réaliser l'aiguillage multibranches, c'est à dire des traitements exclusifs les uns des autres, exécutés suivant les valeurs possibles d'une variable. Sur une structure de type 'CASE OF' on définit uniquement le nom de la variable (dans la partie condition et sur une seule ligne), les valeurs possibles de cette variable sont précisées sur les structures (non élémentaires) de niveau hiérarchique immédiatement inférieur.</p> <p>Dans un écran Dialogue, l'imbrication de structure de type 'CASE OF' n'est pas autorisée.</p> |
| | | DW | <p>Structure de type 'DO WHILE'.</p> <p>Interdite au niveau fonction et au niveau 99. Structure exécutée de façon répétitive tant que son conditionnement est satisfait.</p> |
| | | DU | <p>Structure de type 'DO UNTIL'.</p> <p>Interdite au niveau fonction et au niveau 99. Structure exécutée de façon répétitive jusqu'à ce que son conditionnement soit satisfait. Elle est donc exécutée au moins une fois.</p> <p>Pour ces deux derniers types de structure il convient de veiller à faire évoluer les paramètres du conditionnement pour éviter de boucler.</p> |
| | | DO | <p>Structure de type 'DO' (structure de type boucle).</p> <p>Interdite au niveau fonction et 99.</p> <p>Structure exécutée de façon répétitive entre 2 bornes (précisées dans la zone 'Conditionnement') en fonction d'un pas positif ou négatif. Le pas est facultatif. Les bornes peuvent être soit une valeur, soit un nombre (signé ou non). Elles doivent figurer sur la ligne titre de la sous-fonction, séparées par un espace.</p> <p>Les paramètres doivent être renseignés dans l'ordre suivant: Borne inférieure (positive), borne supérieure et le pas d'incrément de l'indice.</p> <p>DO exige 3 paramètres, dont les 2 premiers sont obligatoires. Tous les 3 doivent paraître sur la première ligne.</p> <p>PACBASE génère automatiquement un indice JffnnR, 'ffnn' représentant le code (sous-)fonction.</p> |
| | | OR | <p>Suite de conditionnement associée aux lignes précé-</p> |

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|---|
| | | | <p>dentes par un 'OU' logique.</p> |
| | | AN | <p>Suite de conditionnement associée aux lignes précédentes par un 'ET' logique.</p> <p>N-B: les parenthèses regroupant des termes dans une condition doivent être indiquées dans le texte de la condition.</p> <p>Dans une Macro-Structure le type de structure ou de condition ne peut pas être paramétré.</p> |
| | | WH | <p>Vous pouvez utiliser des ordres COBOL derrière un ordre EVA ou SEA.</p> <p>Chacun des ordres indique le traitement à effectuer pour une condition particulière.</p> <p>Dans l'exemple suivant, le traitement t1 est effectué lorsque la condition c1 est remplie.</p> <p>EVA ----- ---t1--- 99WH ---c1--- ---t2--- 99WH ---c2--- ---t3--- 99WH ---c3---</p> <p>DIALOGUE -----</p> <p>Les types suivants sont réservés aux modules DIALOGUE et PACBENCH C/S :</p> |
| | | *A | <p>Insertion de la sous-fonction avant une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.</p> |
| | | *P | <p>Insertion de la sous-fonction après une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite. En fonction 20 indiquer le code.</p> |
| | | *R | <p>Remplacement d'une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.</p> <p>Pour plus de détails concernant ces trois types, se reporter au Manuel de Référence Dialogue ou Pacbench C/S, Chapitre 'Emploi du langage structuré', Sous-chapitre 'Traitements spécifiques'.</p> <p>Les deux types suivants sont réservés au module Pacbench C/S (composants serveur uniquement) :</p> |
| | | *C | <p>Insertion ou remplacement des traitements sur serveur ou vue logique. Il ne peut être indiqué que sur un niveau 05.</p> |

| NUM LON | CLASSE VALEUR *B | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE Insertion dans les traitements élémentaires appelés par PERFORM. Pour plus de détails concernant ces deux types, se reporter au Manuel de Référence Pacbench C/S, Chapitre 'Emploi du langage structuré', Sous-chapitre 'Traitements spécifiques'. |
|---------|------------------------|--|
| 10 28 | | CONDITION D'EXECUTION Exprimé sans 'IF', ni 'AND', ni 'OR', ni 'GO TO', ni point. La première ligne d'une condition doit obligatoirement être associée à un niveau hiérarchique explicite ou implicite (05 pour une fonction, 06 à 98 pour une sous-fonction, 99 pour une structure élémentaire à l'intérieur d'une fonction ou sous-fonction). Par contre sur les lignes suite, il ne doit pas être précisé de niveau hiérarchique. Pour une structure de type 'CASE OF' le nom de la va- riable sujet de l'alternative doit être indiqué sur une seule ligne. Pour les structures dépendant d'un 'CASE OF', la con- dition d'exécution représente une valeur possible de la variable sujet du 'CASE OF'. Dans une macro-structure, la condition d'exécution peut être paramétrée. |

6.2. TITRES ET NIVEAUX DES TRAITEMENTS (-TO)

AFFICHAGE DES TITRES ET NIVEAUX DES TRAITEMENTS (-TO)

Comme l'écran "Titres et conditions", l'écran "Titres et niveaux" permet de visualiser la structure générale d'un programme généré.

Il permet une vision légèrement différente des traitements générés par l'indentation des niveaux et des codes des sous-fonctions correspondantes.

A la différence de l'écran "Titres et conditions" les conditions n'apparaissent pas. Pour chaque (sous-) fonction, seul le "type de bloc" ('BL', 'IT', ...) est affiché dans la partie droite de l'écran.

MISE EN OEUVRE

Cet écran est obtenu par le choix :

.P.....TO' pour les Programmes,
.O.....TO' pour les Ecrans.

Cet écran ne peut être mis à jour.

Afin d'éviter d'avoir une image trop détaillée du Programme, il est possible de limiter l'affichage des titres et niveaux à un niveau donné. La commande est alors :

'-TOFFxx<nn' ou '-<nnTOFFxx'.

Valeur par défaut de FFxx = blanc,
Valeur par défaut de nn = 05.

DEBRANCHEMENT SUR UNE (SOUS-)FONCTION AFFICHEE

L'utilisateur a la possibilité de se débrancher sur l'écran '-P' correspondant à une ligne affichée. Pour cela, il doit placer le curseur sur la ligne concernée et utiliser la touche fonction réservée à cet effet (en standard PF10).

CONDITIONS PREREQUISES

Le Programme ou l'Ecran doivent avoir été définis.

Remarque concernant le Module DIALOGUE :

Des différences entre le contenu de l'écran "-TO" et le programme généré peuvent apparaître. Elles tiennent aux raisons suivantes :

Fonction F80 :

Sur l'écran "-TO", l'ordre des sous-fonctions dépend du code des Segments dans le Programme alors que dans le généré, il dépend du code des Segments dans la Bibliothèque.

Pour un Segment appelé sur l'écran "-CS", d'organisation différente de 'U' et non utilisé en affichage ou en réception, l'affichage est quand même simulé sur l'écran "-TO".

Fonction F81 :

Le contrôle de numéricité et le contrôle de date étant considérés comme toujours générés, les sous-fonctions F8110 et F8120 sont toujours affichées sur l'écran "-TO". Or, la sous-fonction F8110 n'est générée que si des Rubriques de nature variable et de format numérique sont appelées. Quant à la sous-fonction F8120 elle n'est générée qu'en cas d'appel de Rubriques de nature variable et de format date.

TITRES, CONDITIONS ET NIVEAUX DES TRAITEMENTS
 TITRES ET NIVEAUX DES TRAITEMENTS (-TO)

6
 2

```

-----
!
!           VALIDATION MATERIEL IBM                       PO*AG.LILI.IBM.847 !
!
! TRAITEMENTS PROGRAMME FO20MA MISE A JOUR FOURNISSEURS !
! . FOSF . TITRE . NIVEAU . TY .SOURCE.BIBLI !
! ..... !
! . 01 . INITIALISATIONS . 05 . BL . . !
! . 05 . LECTURE FICH. SEQ. SS RUPTURE . 05 . BL . . !
! . 05FB . . . . .AAMA11.*VAL !
! . 10 . LECTURE FICH. SEQ. A RUPTURE . 05 . BL . . !
! . 10FB . . . . .AAMA11.*VAL !
! . 19 . VIDAGE DE LA TABLE . 05 . IT . .*VAL !
! . 20 . FIN DE TRAITEMENT . 05 . IT . . !
! . 22 . CALCUL DERNIERS ENREG. . 05 . BL . . !
! . 24 . . . . .AAMA11.*VAL !
! . 26 . CALCUL RUPTURES TOTALES . 05 . BL .AAMA11.*VAL !
! . 30 . CALCUL VARIABLES CONTROLE . 05 . IT . . !
! . 33 . CONTROLE IDENTIFICATION . 05 . IT . . !
! . 36 . CONTROLE ENREGISTR. EN DOUBLE . 05 . BL . . !
! . 39 . DETERMINATION PRESENCE RUBR. . 05 . IT . . !
! . 42 . CONTROLE DE STRUCTURE ENREG. . 05 . IT . . !
! . 45 . CONTROLE CONTENU RUBRIQUES . 05 . IT . . !
! . 46 . AUTRES CONTROLES . 05 . BL . .*VAL !
! . 51 . CONTROLE PRESENCE ENREG. . 05 . IT . . !
! SIMULATION LIMITEE AU NIVEAU: 05 !
! O: C1 CH: -TO !
!
-----

```

7. REPERTOIRE D'ACCES

7.1. ACCES AUX ECRANS

CHOIX : PROGRAMMES

LISTE DES PROGRAMMES

LCPaaaaaa

Liste des Programmes par code (à partir du Programme 'aaaaaa').

LEPaaaaaaaa

Liste des Programmes par nom externe (à partir du nom externe 'aaaaaaaa').

LTPtPaaaaaa

Liste des Programmes par type (à partir du type 't' et du Programme 'aaaaaa').

DESCRIPTION DU PROGRAMME 'aaaaaa'

Paaaaaa (MAJ)

Définition du Programme 'aaaaaa'.

PaaaaaaGbbb (MAJ)

Documentation du Programme 'aaaaaa' (à partir de la ligne 'bbb').

PaaaaaaATbbbbbb

Textes associés au Programme 'aaaaaa' (à partir du Texte 'bbbbbb').

PaaaaaaX

Utilisations du Programme 'aaaaaa'.

PaaaaaaXVbbbbbb

Utilisations du Programme 'aaaaaa' dans les Rapports (G.D.P.) (à partir du Rapport 'bbbbbb').

PaaaaaaXPbbbbbb (MAJ)

Utilisations du Programme 'aaaaaa' dans les Programmes (à partir du Programme 'bbbbbb').

PaaaaaaXObbbbbbb

Utilisations du Programme 'aaaaaa' dans les Ecrans (à partir de l'Ecran 'bbbbbb').

PaaaaaaXQbbbbbb

Utilisations du Programme 'aaaaaa' dans les Relations Utilisateur (à partir de la R.U. 'bbbbbb').

PaaaaaaCDbb (MAJ)

Appels de Structures de Données dans le Programme 'aaaaaa' (à partir de la Structure de Données 'bb').

PaaaaaaHCDbb (MAJ)

"Zoom" sur l'appel de la Structure de Données 'bb' dans le Programme 'aaaaaa'.

PaaaaaaCPbbbbbb (MAJ)

Appels de Macro-Structures dans le Programme 'aaaaaa' (à partir de la Macro-Structure 'bbbbbb').

PaaaaaaBbbccddd (MAJ)

Modification du début du Programme 'aaaaaa' (à partir de la section 'bb' paragraphe 'cc', ligne 'ddd').

PaaaaaaWbbccc (MAJ)

Descriptif des zones de travail du Programme 'aaaaaa' (à partir de l'emplacement 'bb' et du numéro de ligne 'ccc').

PaaaaaaPbbccddd (MAJ)

Descriptif de traitement structuré du Programme 'aaaaaa' (à partir de la fonction 'bb', sous-fonction 'cc' numéro de ligne 'ddd').

PaaaaaaPGbbccddd (MAJ)

Descriptif de traitement structuré du Programme 'aaaaaa' (à partir de la fonction 'bb', sous-fonction 'cc' numéro de ligne 'ddd') et affichage des titres des traitements générés.

Paaaaaa9bbbbbb (MAJ)

Descriptif de traitement <COBOL pur> du Programme 'aaaaaa' (à partir de la ligne 'bbbbbb').

Paaaaaa8bbccnnn (MAJ)

Descriptif des traitements du Programme 'aaaaaa' (à partir de la fonction 'bb', sous-fonction 'cc' et numéro de ligne 'nnn'), réservé à la maintenance de Programmes (versions antérieures à la version 6 du Système.

PaaaaaaTCbbcc (MAJ)

Descriptif des titres et conditions des traitements automatiques et spécifiques (à partir de la fonction 'bb', sous-fonction 'cc').

PaaaaaaTCbbcc<nn ou Paaaaaa<nnTCbbcc (MAJ)

Descriptif des titres et conditions des traitements automatiques et spécifiques jusqu'au niveau 'nn' (à partir de la fonction 'bb', sous-fonction 'cc').

PaaaaaaTObbcc

Descriptif des titres des traitements automatiques et spécifiques (à partir de la fonction 'bb', sous-fonction 'cc') avec indentation des niveaux.

PaaaaaaTObbcc<nn ou Paaaaaa<nnTObbcc

Descriptif des titres des traitements automatiques et spécifiques jusqu'au niveau 'nn' (à partir de la fonction 'bb', sous-fonction 'cc').

NB : 'Paaaaaa' peut être remplacé par '-' après la première transaction de type 'Paaaaaa'.

Tous les éléments entre parenthèses sont facultatifs.

```

-----
!
!           SUIVI PERMIS DE CONSTRUIRE           P0*DOC.LURE.JIA.893 !
!
! LISTE DES PROGRAMMES
!
! PROGRA MEMBRE NOM DU PROGRAMME OU DU MODULE V N AV AP PROGR.ID TNP TYPE BIBL !
! AAAC30 AAAC30 LECTURE DIRECTE FICHER CICS N          AAAC30 PM P *CEN !
! AAAC40 AAAC40 SQUELETTE C.I.C.S. N          AAAC40 PM P *CEN !
! AAAC50 AAAC50 AFFICHAGE ERREURS SOUS CICS N          AAAC50 PM P *CEN !
! AAAC70 AAAC70 MISE A JOUR SOUS CICS N          AAAC70 PM P *CEN !
! AAADOC AAADOC Appel programme de doc. N          AAADOC PB P 0353 !
! AAC10 AAC10 CADRAGE D'UNE ZONE SAISIE N          AAC10 PB P *CEN !
! AAC11 AAC11 APPEL CADRAGE D'UNE ZONE N          AAC11 PM P *CEN !
! AADA10 AADA10 CONTROLE DE DATE N N          AADA10 PB M *CEN !
! AADA11 AADA11 MEMORISATION DE L'ERREUR N N          AADA11 PB M *CEN !
! AAED12 AAED12 TRAITEMENTS POUR EDITIONS N          AAED12 PB M 0136 !
! AAER10 AAER10 EDITION DES ERREURS N          AAER10 PB M *CEN !
! AAER12 AAER12 REPRISE ETAT INIT NIVEAU MAJ N          AAER12 PB M *CEN !
! AAJIAE AAJIAE TRAITEMENTS COMMUNS P.C N          AAJIAE PB M 0126 !
! AAMOIS AAMOIS table des libelles des mois N N          AAMOIS PB M *CEN !
! AAPR20 AAPR20 DISPLAY DEBUT ET FIN DE PGM N N          AAPR20 PB M *CEN !
! AARECH AARECH TRAITEMENTS RECHERCHE DE P.C N          AARECH PB M 0138 !
! AATR10 AATR10 TRI INPUT - OUTPUT N          AATR10 PB M *CEN !
! AATR20 AATR20 TRI INPUT PROCEDURE N          AATR20 PB M *CEN !
!
! O: C1 CH: LCP
!
-----

```

```

-----
!
!           SUIVI PERMIS DE CONSTRUIRE           P0*DOC.LURE.JIA.893 !
!
! TEXTES VENTILES SUR LE PROGRAMME JIPED1
!
! ---- TEXTE BBA -----RUBRI.
! DD 100 K          ** EDITIONS STATISTIQUES **
! DD 102 I RED1 RED2 . . . . .
! DD 104 I PJIPED1PJIPED2. . . . .
! DD 110 3 POUR L'ENSEMBLE DES DOSSIERS TRAITES DANS LA DDE SUR LE
! DD 120 MINI-ORDINATEUR, LES EDITIONS SUIVANTES SERONT PREVUES : COCOMM
! DD 130 2 - PAR ANNEE DE DEPOT, COMMUNE ET NATURE D'ARRETE : NATAR
! DD 140 NOMBRE ET POURCENTAGE DE DOSSIERS.
! DD 150 2 - PAR ANNEE DE DEPOT, MOIS ET COMMUNE :
! DD 140 NOMBRE ET POURCENTAGE DE DOSSIERS.
! ZB 100 L TITRE DU CHAPITRE
! ZB 110 *
! ZB 115 5
! ZB 120 9          OBJECTIFS DE GESTION
! ZB 130 -----
! ZD 100 L TITRE DU CHAPITRE
! ZD 110 *
! ZD 115 4
! ZD 120 9          TRAITEMENTS RELATIFS A LA PROCEDURE
!
! O: C1 CH: -AT
!
-----

```

7.2. OPTIONS D'AFFICHAGE EN T.P.

OPTIONS D'AFFICHAGE EN TP.

P.....CP :

C1 : Affiche les lignes d'appel de Macro-Structures dans le Programme.

C2 : Affiche le numéro de session où la ligne a été mise à jour.

P.....B, -W, -P, -8 :

C1 : Affiche le format de saisie des informations.

C2 : Affiche les informations concernant l'origine des lignes.

Pour les lignes obtenues dans le Programme par l'appel d'une Macro-Structure, le code de celle-ci apparaît dans la zone SOURCE.

Pour les lignes appartenant à la Bibliothèque dans laquelle on travaille, la zone BIBLI indique le numéro de session où la ligne a été mise à jour. Pour les lignes appartenant à d'autres Bibliothèques, elle indique leurs codes.

Cette option ne permet pas la mise à jour.

C3 : Disponible sur les écrans -W uniquement. Affiche le format sélectionné (I-E-S) pour la Rubrique.

Cette option ne permet pas la mise à jour.

7.3. CODES ACTION EN T.P.

CODES ACTIONS

Sur la ligne de définition :

C= création de la ligne dans la bibliothèque,

M= modification de la ligne,

A= annulation de la ligne. Elle n'est possible que s'il n'existe aucune ligne de description,

?= demande de documentation.

Sur les lignes de description :

C= création de la ligne dans la bibliothèque,

M= modification de la ligne,

A= annulation de la ligne,

B= annulation multiple,

R= duplication de ligne,

I= insertion de ligne,

T= transfert de ligne,

G= transfert multiple,

L= fin d'annulation ou de transfert multiples,

?= demande de documentation.

Pour plus de détails sur le Code Action, voir le Manuel d'UTILISATION.

7.4. MODE D'ACCES EN BATCH

DEFINITION DE PROGRAMME

En batch un Programme est défini à l'aide d'un bordereau 0 (zéro).

Codes action :

- C = création de la ligne dans la bibliothèque.
- M = modification de la ligne.
- Blanc = création ou modification, suivant l'état de la bibliothèque.
- X = création ou modification avec insertion du caractère '&'.
- B = annulation du Programme: Définition '0' et Documentation Générale (V3), ainsi que toutes ses lignes de description (7, 1, P, M, D, 9, 8 et FC).
Les lignes de texte ne sont pas supprimées.

APPEL DE STRUCTURES DE DONNEES

Une Structure de Données est appelée à l'aide d'un bordereau 1.

Cette ligne doit toujours être précédée d'une ligne de définition de Programme car elle ne comporte pas le code Programme.

Code Action :

- C = création de la ligne dans la bibliothèque.
- M = modification de la ligne.
- Blanc = création ou modification, suivant état de la bibliothèque.
- X = création ou modification avec insertion du caractère '&'.
- A = annulation de la ligne ou de toutes les lignes concernant un fichier si le code est associé à la première d'entre elles.
- B = annulation de plusieurs lignes à partir de la ligne indiquée incluse.
- R = fin d'annulation multiple au delà de cette ligne.
Si aucune ligne de code R ne suit une ligne de code B, l'annulation s'arrête :

. à la fin de la section si le code paragraphe est absent,

. à la fin du paragraphe si le code paragraphe est indiqué.

MODIFICATION DU DEBUT DE PROGRAMME

Le début de Programme est modifié à l'aide du bordereau D.

Cette ligne doit toujours être précédée d'une ligne de définition de Programme car elle ne comporte pas le code Programme.

Codes Action :

C = création de la ligne dans la bibliothèque.

M = modification de la ligne.

Blanc = création ou modification, suivant état de la bibliothèque.

X = création ou modification avec insertion du caractère '&', sans transformation des minuscules en majuscules dans la zone 'Instruction à insérer'.

A = annulation de la ligne.

B = annulation multiple
à partir de la ligne indiquée incluse.

R = fin d'annulation multiple au-delà de cette ligne.
Si aucune ligne de code R ne suit une ligne de code B,
l'annulation s'arrête :

. à la fin de la section si le code paragraphe est absent,

. à la fin du paragraphe si le code paragraphe est indiqué.

ZONES DE TRAVAIL ET DE LIEN

Les zones de travail et de lien sont décrites à l'aide du bordereau 7.

Cette ligne doit toujours être précédée d'une ligne de définition de programme car elle ne comporte pas le code programme.

Codes Action :

C = création de la ligne dans la bibliothèque.

M = modification de la ligne.

Blanc = création ou modification, suivant état de la bibliothèque.

X = création ou modification avec insertion du caractère '&' et pas de transformation des minuscules en majuscules.

A = annulation de la ligne.

B = annulation de plusieurs lignes dans le programme à partir de la ligne indiquée incluse.

R = fin d'annulation multiple au-delà de cette ligne.
Si aucune ligne de type R ne suit une ligne de type B, l'annulation s'arrête à la fin des lignes '7'.

DESCRIPTION DE TRAITEMENTS

Les traitements sont décrits à l'aide du bordereau P.

Cette ligne doit toujours être précédée d'une ligne de définition de Programme car elle ne comporte pas le code Programme.

Codes Action :

C = création de la ligne dans la Bibliothèque.

M = modification de la ligne.

Blanc = création ou modification, suivant l'état de la Bibliothèque.

X = création ou modification avec insertion du caractère '&' et pas de transformation des minuscules en majuscules.

A = annulation de la ligne.

B = annulation de plusieurs lignes dans le Programme à partir du numéro de ligne indiqué inclus.

R = fin d'annulation multiple au-delà de cette ligne.
Si aucune ligne de code R ne suit une ligne de code B l'annulation s'arrête :

. à la fin de la fonction si le code sous-fonction est absent,

. à la fin de la sous-fonction si le code sous-fonction est présent.

APPEL DE MACRO-STRUCTURES

Les Macro-Structures sont appelées à l'aide du bordereau M.

Cette ligne doit toujours être précédée d'une ligne de définition de Programme car elle ne comporte pas le code Programme.

Codes Action :

- C = création de la ligne dans la Bibliothèque.
- M = modification de la ligne.
- Blanc = création ou modification, suivant l'état de la Bibliothèque.
- X = création ou modification avec insertion du caractère '&' et pas de transformation des minuscules en majuscules.
- A = annulation de la ligne.
- B = annulation de l'incorporation d'une Macro-Structure dans un programme, à partir du numéro de ligne indiqué inclus.
- R = fin d'annulation multiple au-delà de cette ligne.
Si aucune ligne de type R ne suit une ligne de type B, l'annulation s'arrête à la dernière ligne d'appel de la Macro-Structure.

7.5. EDITION ET GENERATION

COMMANDES D'EDITION ET GENERATION DES PROGRAMMES

LCP : Liste des Programmes :

- . option C1 : édition de la ligne de définition de chaque Programme, sans les mots-clés.
- . option C2 : avec les mots-clés.

LKP : Liste des programmes par mots-clés. Cette commande doit être accompagnée d'une ligne suite où seront indiqués les mots-clés servant à la sélection (Voir le Manuel d'utilisation).

DCP : Description du Programme dont le code est indiqué dans la zone Entité.
Description de tous les Programmes si la zone n'est pas renseignée :

- . option C1 : édition de toutes les lignes de description du Programme sans les Textes ventilés.
- . option C2 : avec les Textes ventilés.

GCP : Génération et description du Programme dont le code est obligatoirement indiqué. Mêmes options d'édition que pour DCP.

FLP : Contrôle du flot des programmes. L'utilisateur peut préciser l'environnement (PEI), des options cartes de contrôle et éventuellement des paramètres.

8. DESCRIPTION D'UN PROGRAMME GENERE

8.1. INTRODUCTION

DESCRIPTION D'UN PROGRAMME GENERE

Ce chapitre fournit un exemple de la partie générée automatiquement d'un Programme de nature 'P'. Il contient donc les trois premières divisions du COBOL, la procédure, écrite en lignes 'P', ne posant a priori pas de problèmes de compréhension des instructions COBOL générées.

Les principales caractéristiques des lignes d'appel de S.D. du Programme sont résumées dans les deux tableaux suivants.

Lignes d'appel de Structures de Données :

```

-----
! SD  BI  OAMOU  U  SE  SELECT.  T  N  ZONE  COMPL  CLE  !
-----
! KL  CD  SSVOU  S  LK                1                !
! LA  CI  VRFRD  C                1  LA00-STATUS  CLECI  !
! LB  MO  ISFID  T                1                MOIS  !
! LK  CD  SSFTU  C  KL                1                !
! TR  CD  VSFID  C                1  TR00-STATUS  !
! WK  CD  WSFOU  D          *102030  2  2                !
-----

```

Appels en lignes '-W' :

```

-----
! N.LIG  SD  BI  SELECT.  F  T  N  SOURCE  !
-----
! BB010  WR  WR  02                I  2  2  AADA10  !
! 20010  WG  WG  99                I    1                !
! 20020  CC  CD                I  4  3                !
! 30030  KA  CI                I    2                !
-----

```

DESCRIPTION D'UN PROGRAMME GENERE
IDENTIFICATION DIVISION

PAGE

139

8
2

8.2. IDENTIFICATION DIVISION

IDENTIFICATION DIVISION

L'utilisateur a la possibilité de modifier cette division (-B).

IDENTIFICATION DIVISION.
PROGRAM-ID. PJJPLS.
AUTHOR. COMMANDES.
DATE-COMPILED. 12/04/91.

PJJPLS
PJJPLS
PJJPLS

8.3. ENVIRONMENT DIVISION

ENVIRONMENT DIVISION

Cette division est entièrement adaptée à la variante du COBOL demandée pour le Programme (IBM OS dans l'exemple).

De manière générale :

- . Trois types d'organisation de fichier sont traités :
 - . organisation séquentielle
 - . organisation indexée
 - . organisation 'V' séquentielle ou indexée (VSAM sur IBM, UFAS sur CII-HB, etc.)
- . Trois types d'accès sont traités :
 - . accès séquentiel
 - . accès dynamique (organisation 'V' indexée seulement)
 - . accès direct.

Dans ce dernier cas la NOMINAL KEY (ou SYMBOLIC KEY) générée est toujours de la forme 1-XXNN-CORUB ou XX, NN, CORUB ont été définis par l'utilisateur dans les appels de S.D. En effet, cette clé figure généralement dans un fichier en entrée sinon il appartient à l'utilisateur de la définir et de la gérer.

Remarque : Cette partie du Programme peut être modifiée par des lignes '-B'.

DESCRIPTION D'UN PROGRAMME GENERE
ENVIRONMENT DIVISION

PAGE

141

8
3

| | |
|---|--------|
| ENVIRONMENT DIVISION. | PJJPLS |
| CONFIGURATION SECTION. | PJJPLS |
| SOURCE-COMPUTER. IBM-370. | PJJPLS |
| OBJECT-COMPUTER. IBM-370. | PJJPLS |
| INPUT-OUTPUT SECTION. | PJJPLS |
| FILE-CONTROL. | PJJPLS |
| SELECT KL-FICHER ASSIGN UT-S-SOR01. | PJJPLS |
| SELECT LA-FICHER ASSIGN TO ENT02 | PJJPLS |
| ORGANIZATION INDEXED | PJJPLS |
| ACCESS IS RANDOM | PJJPLS |
| FILE STATUS IS 1-LA00-STATUS | PJJPLS |
| RECORD KEY IS LA00-NOCL. | PJJPLS |
| SELECT LB-FICHER ASSIGN DA-I-LB | PJJPLS |
| RECORD KEY IS LB00-MOIS. | PJJPLS |
| SELECT LK-FICHER ASSIGN UT-S-TRI01. | PJJPLS |
| SELECT TR-FICHER ASSIGN TO ENT01 | PJJPLS |
| ORGANIZATION INDEXED | PJJPLS |
| FILE STATUS IS 1-TR00-STATUS | PJJPLS |
| RECORD KEY IS TR00-CLECD. | PJJPLS |

8.4. DATA DIVISION FILE SECTION

DATA DIVISION

L'utilisateur ne peut en aucun cas modifier cette partie du Programme, si ce n'est par la description des fichiers.

Tous les fichiers du Programme sauf ceux dont l'organisation est X, W, L ou D apparaissent en FILE SECTION. Ils sont décrits à un niveau de détail plus ou moins grand selon leur utilisation et le type de description demandé.

Chaque enregistrement décrit apparaît sous la forme XXNN où XX est le code de la Structure de Données dans le Programme et NN le code de l'enregistrement décrit.

Chaque Rubrique décrite apparaît sous la forme XXNN-CORUB avec sa PICTURE ou sous-définie si elle est déclarée comme groupe dans la Structure de Données.

Les fichiers pour lesquels il a été demandé une description étalée n'ont qu'un seul enregistrement COBOL XX00 qui englobe la partie commune et les parties spécifiques décrites dans la base.

Les fichiers en entrée, en entrée-sortie, les fichiers divers en sortie (utilisation D) sont décrits complètement.

Pour les fichiers en sortie chaînés à des fichiers en entrée le détail des Rubriques n'apparaît pas à ce niveau.

DESCRIPTION D'UN PROGRAMME GENERALE
 DATA DIVISION FILE SECTION

| | | | | |
|----------------|-------------|------------------------|-------------------|--------|
| DATA DIVISION. | | | | PJJPLS |
| FILE SECTION. | | | | PJJPLS |
| FD | | KL-FICHER | | PJJPLS |
| | BLOCK | | 0000 RECORDS | PJJPLS |
| | RECORDING | V | | PJJPLS |
| | DATA RECORD | | | PJJPLS |
| | | KL00 | | PJJPLS |
| | | KL05 | | PJJPLS |
| | | KL10 | | PJJPLS |
| | | KL20 | | PJJPLS |
| | | KL30 | | PJJPLS |
| | | LABEL RECORD STANDARD. | | PJJPLS |
| 01 | | KL00. | | PJJPLS |
| | 10 | FILLER | PICTURE X(00166). | PJJPLS |
| 01 | | KL05. | | PJJPLS |
| | 10 | FILLER | PICTURE X(00166). | PJJPLS |
| 01 | | KL10. | | PJJPLS |
| | 10 | FILLER | PICTURE X(00148). | PJJPLS |
| 01 | | KL20. | | PJJPLS |
| | 10 | FILLER | PICTURE X(00010). | PJJPLS |
| 01 | | KL30. | | PJJPLS |
| | 10 | FILLER | PICTURE X(00014). | PJJPLS |
| FD | | LA-FICHER | | PJJPLS |
| | BLOCK | | 0000 RECORDS | PJJPLS |
| | DATA RECORD | | | PJJPLS |
| | | LA00 | | PJJPLS |
| | | LABEL RECORD STANDARD. | | PJJPLS |
| 01 | | LA00. | | PJJPLS |
| | 10 | LA00-CLECI. | | PJJPLS |
| | 11 | LA00-NOCL. | | PJJPLS |
| | 12 | LA00-NOCL11 | PICTURE X. | PJJPLS |
| | 12 | LA00-NOCL12 | PICTURE XX. | PJJPLS |
| | 12 | LA00-NOCL2 | PICTURE XX. | PJJPLS |
| | 10 | LA00-NOMCL | PICTURE X(20). | PJJPLS |
| | 10 | LA00-ADRES | PICTURE X(43). | PJJPLS |
| | 10 | LA00-NUDEP | PICTURE XXX. | PJJPLS |
| | 10 | LA00-LIDEP | PICTURE X(24). | PJJPLS |
| | 10 | LA00-NUREG | PICTURE XXX. | PJJPLS |
| | 10 | LA00-LIREG | PICTURE X(24). | PJJPLS |
| FD | | LB-FICHER | | PJJPLS |
| | BLOCK | | 00001 RECORDS | PJJPLS |
| | DATA RECORD | | | PJJPLS |
| | | LB00 | | PJJPLS |
| | | LB10 | | PJJPLS |
| | | LB20 | | PJJPLS |
| | | LABEL RECORD STANDARD. | | PJJPLS |
| 01 | | LB00. | | PJJPLS |
| | 05 | LB00-00. | | PJJPLS |
| | 10 | LB00-ANNUL | PICTURE X. | PJJPLS |
| | 10 | LB00-MOIS | PICTURE 99. | PJJPLS |
| | 10 | LB00-LMOIS | PICTURE X(9). | PJJPLS |
| | 10 | LB00-FILLER | PICTURE X(68). | PJJPLS |
| | 10 | LB00-ACTIO | PICTURE X. | PJJPLS |
| | 10 | LB00-FLM | PICTURE 999. | PJJPLS |
| | 10 | LB00-FLDDEP | PICTURE X(6). | PJJPLS |
| | 10 | LB00-STRUCT | PICTURE X. | PJJPLS |
| | 10 | LB00-PANFI | PICTURE X. | PJJPLS |
| | 10 | LB00-PANL | PICTURE X(12). | PJJPLS |
| | 05 | LB00-SUITE. | | PJJPLS |
| | 15 | FILLER | PICTURE X(00050). | PJJPLS |
| 01 | | LB10. | | PJJPLS |
| | 10 | FILLER | PICTURE X(00104). | PJJPLS |
| | 10 | LB10-FLDCAN | PICTURE X(6). | PJJPLS |
| | 10 | LB10-FLCADE | PICTURE X(3). | PJJPLS |
| | 10 | LB10-FLCAAR | PICTURE X(3). | PJJPLS |
| | 10 | LB10-FLTDEP | PICTURE X(6). | PJJPLS |
| | 10 | LB10-FLTARR | PICTURE X(6). | PJJPLS |
| | 10 | LB10-ACMSER | PICTURE 9(6). | PJJPLS |
| | 10 | LB10-FLCFS | PICTURE X. | PJJPLS |
| | 10 | LB10-FLQSTP | PICTURE 9. | PJJPLS |
| | 10 | LB10-FILLER | PICTURE X(18). | PJJPLS |
| 01 | | LB20. | | PJJPLS |
| | 10 | FILLER | PICTURE X(00104). | PJJPLS |
| | 10 | LB20-PANTTL | PICTURE XXX. | PJJPLS |
| | 10 | LB20-PAMPHH | PICTURE X(10). | PJJPLS |
| | 10 | LB20-PAMPHW | PICTURE X(10). | PJJPLS |
| | 10 | LB20-RECCLA | PICTURE X. | PJJPLS |
| | 10 | LB20-RECSMK | PICTURE X. | PJJPLS |

DESCRIPTION D'UN PROGRAMME GENERE
 DATA DIVISION FILE SECTION

| | | | | |
|----|----|------------------------|-----------|--------|
| | 10 | LB20-REDCAN PICTURE | X(6). | PJJPLS |
| | 10 | LB20-FILLER PICTURE | X(19). | PJJPLS |
| SD | | LK-FICHER | | PJJPLS |
| | | DATA RECORD | | PJJPLS |
| | | LK00 | | PJJPLS |
| | | LK05 | | PJJPLS |
| | | LK10 | | PJJPLS |
| | | LK20 | | PJJPLS |
| | | LK30. | | PJJPLS |
| 01 | | LK00. | | PJJPLS |
| | 05 | LK00-00. | | PJJPLS |
| | 10 | LK00-CLECD. | | PJJPLS |
| | 11 | LK00-COCARA PICTURE | X. | PJJPLS |
| | 11 | LK00-NUCOM PICTURE | 9(5). | PJJPLS |
| | 11 | LK00-FOURNI PICTURE | X(3). | PJJPLS |
| | 05 | LK00-SUITE. | | PJJPLS |
| | 15 | FILLER PICTURE | X(00157). | PJJPLS |
| 01 | | LK05. | | PJJPLS |
| | 10 | FILLER PICTURE | X(00009). | PJJPLS |
| | 10 | LK05-NUCLIE PICTURE | 9(8). | PJJPLS |
| | 10 | LK05-DATE PICTURE | X(6). | PJJPLS |
| | 10 | LK05-RELEA PICTURE | X(3). | PJJPLS |
| | 10 | LK05-REFCLI PICTURE | X(30). | PJJPLS |
| | 10 | LK05-RUE PICTURE | X(40). | PJJPLS |
| | 10 | LK05-COPOS PICTURE | X(5). | PJJPLS |
| | 10 | LK05-VILLE PICTURE | X(20). | PJJPLS |
| | 10 | LK05-CORRES PICTURE | X(25). | PJJPLS |
| | 10 | LK05-REMIS PICTURE | S9(4)V99. | PJJPLS |
| | 10 | LK05-MATE PICTURE | X(8). | PJJPLS |
| | 10 | LK05-LANGU PICTURE | X. | PJJPLS |
| | 10 | LK05-FILLER PICTURE | X(5). | PJJPLS |
| 01 | | LK10. | | PJJPLS |
| | 10 | FILLER PICTURE | X(00009). | PJJPLS |
| | 10 | LK10-QTMAC PICTURE | 99. | PJJPLS |
| | 10 | LK10-QTMAL PICTURE | 99. | PJJPLS |
| | 10 | LK10-INFOR PICTURE | X(35). | PJJPLS |
| | 10 | LK10-RFI PICTURE | X(100). | PJJPLS |
| | 10 | FILLER PICTURE | X(00018). | PJJPLS |
| 01 | | LK20. | | PJJPLS |
| | 10 | FILLER PICTURE | X(00009). | PJJPLS |
| | 10 | LK20-EDIT PICTURE | X. | PJJPLS |
| | 10 | FILLER PICTURE | X(00156). | PJJPLS |
| 01 | | LK30. | | PJJPLS |
| | 10 | FILLER PICTURE | X(00009). | PJJPLS |
| | 10 | LK30-NUCOM PICTURE | 9(5). | PJJPLS |
| | 10 | FILLER PICTURE | X(00152). | PJJPLS |
| FD | | TR-FICHER | | PJJPLS |
| | | BLOCK 00000 RECORDS | | PJJPLS |
| | | DATA RECORD | | PJJPLS |
| | | TR00 | | PJJPLS |
| | | TR05 | | PJJPLS |
| | | TR10 | | PJJPLS |
| | | TR20 | | PJJPLS |
| | | TR30 | | PJJPLS |
| | | LABEL RECORD STANDARD. | | PJJPLS |
| 01 | | TR00. | | PJJPLS |
| | 05 | TR00-00. | | PJJPLS |
| | 10 | TR00-CLECD. | | PJJPLS |
| | 11 | TR00-COCARA PICTURE | X. | PJJPLS |
| | 11 | TR00-NUCOM PICTURE | 9(5). | PJJPLS |
| | 11 | TR00-FOURNI PICTURE | X(3). | PJJPLS |
| | 05 | TR00-SUITE. | | PJJPLS |
| | 15 | FILLER PICTURE | X(00157). | PJJPLS |
| 01 | | TR05. | | PJJPLS |
| | 10 | FILLER PICTURE | X(00009). | PJJPLS |
| | 10 | TR05-NUCLIE PICTURE | 9(8). | PJJPLS |
| | 10 | TR05-DATE PICTURE | X(6). | PJJPLS |
| | 10 | TR05-RELEA PICTURE | X(3). | PJJPLS |
| | 10 | TR05-REFCLI PICTURE | X(30). | PJJPLS |
| | 10 | TR05-RUE PICTURE | X(40). | PJJPLS |
| | 10 | TR05-COPOS PICTURE | X(5). | PJJPLS |
| | 10 | TR05-VILLE PICTURE | X(20). | PJJPLS |
| | 10 | TR05-CORRES PICTURE | X(25). | PJJPLS |
| | 10 | TR05-REMIS PICTURE | S9(4)V99. | PJJPLS |
| | 10 | TR05-MATE PICTURE | X(8). | PJJPLS |
| | 10 | TR05-LANGU PICTURE | X. | PJJPLS |
| | 10 | TR05-FILLER PICTURE | X(5). | PJJPLS |
| 01 | | TR10. | | PJJPLS |

DESCRIPTION D'UN PROGRAMME GENERE
DATA DIVISION FILE SECTION

PAGE

145

8
4

| | | | | | |
|----|----|------------|---------|-----------|--------|
| | 10 | FILLER | PICTURE | X(00009). | PJJPLS |
| | 10 | TR10-QTMAC | PICTURE | 99. | PJJPLS |
| | 10 | TR10-QTMAL | PICTURE | 99. | PJJPLS |
| | 10 | TR10-INFOR | PICTURE | X(35). | PJJPLS |
| | 10 | TR10-RFI | PICTURE | X(100). | PJJPLS |
| | 10 | FILLER | PICTURE | X(00018). | PJJPLS |
| 01 | | TR20. | | | PJJPLS |
| | 10 | FILLER | PICTURE | X(00009). | PJJPLS |
| | 10 | TR20-EDIT | PICTURE | X. | PJJPLS |
| | 10 | FILLER | PICTURE | X(00156). | PJJPLS |
| 01 | | TR30. | | | PJJPLS |
| | 10 | FILLER | PICTURE | X(00009). | PJJPLS |
| | 10 | TR30-NUCOM | PICTURE | 9(5). | PJJPLS |
| | 10 | FILLER | PICTURE | X(00152). | PJJPLS |

8.5. DEBUT DE WORKING-STORAGE SECTION

DEBUT DE LA WORKING-STORAGE SECTION

Sont générées en tête de la WORKING-STORAGE SECTION les descriptions des Structures de Données en organisation W ou organisation L et de code emplacement alphabétique. Tous les types de description sont possibles dans cette zone.

On peut également insérer, par ligne '-W', des niveaux complémentaires, soit entre les Structures de Données, soit entre les enregistrements d'une même Structure.

DEBUT-WSS sera généré dans tout programme après ces descriptions.

La variable 'IK' n'apparaît que dans les Programmes écrits en langage structuré ('-P').

CONSTANTES-PAC :

Ces constantes sont toujours générées. Elles incluent :
le numéro de la session au cours de laquelle le Programme a été généré (NUGNA), le code de la bibliothèque (APPLI), la date de la génération (DATGN), le code du Programme en Bibliothèque PACBASE (PROGR) et le code de l'utilisateur ayant demandé la génération.

La variable 'DATCE' est composée d'une zone 'CENTUR' contenant la valeur du siècle courant, et d'une zone date (DATOR) non initialisée dans laquelle l'utilisateur pourra stocker la date du traitement sous la forme an-mois-jour (DATOA-DATOM-DATOJ).

Remarque : si vous utilisez un opérateur de récupération de la date du jour (ADT ou ADC), et si l'année est inférieure à '61', la zone 'CENTUR' est automatiquement initialisée à '20'.

DESCRIPTION D'UN PROGRAMME GENERE
DEBUT DE WORKING-STORAGE SECTION

8
5

```

WORKING-STORAGE SECTION.
*SD: WR BIB: WR SEL: 02_____ FORM: I DESC: 2 NIV: 2 ORG: _ SS: _ 7BB010
01      WR00.                                PJJPLS
      02      WR02.                                PJJPLS
      10      WR02-DAT1.                          PJJPLS
      11      WR02-DAT11 PICTURE XX              PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-DAT12 PICTURE XX              PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-DAT13 PICTURE XX              PJJPLS
      VALUE                                     SPACE.    PJJPLS
      10      WR02-DAT113 PICTURE XX             PJJPLS
      VALUE                                     SPACE.    PJJPLS
      10      WR02-AMJ.                            PJJPLS
      11      WR02-AMJA.                            PJJPLS
      12      WR02-AMJA9 PICTURE 99              PJJPLS
      VALUE                                     ZERO.      PJJPLS
      11      WR02-AMJM PICTURE XX               PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-AMJJ PICTURE XX               PJJPLS
      VALUE                                     SPACE.    PJJPLS
      10      WR02-BIS PICTURE 9                 PJJPLS
      VALUE                                     ZERO.      PJJPLS
      10      WR02-DHORDI.                          PJJPLS
      11      WR02-DORDI.                          PJJPLS
      12      WR02-DORDIA PICTURE XX             PJJPLS
      VALUE                                     SPACE.    PJJPLS
      12      WR02-DORDIM PICTURE XX             PJJPLS
      VALUE                                     SPACE.    PJJPLS
      12      WR02-DORDIJ PICTURE XX            PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-HORDI PICTURE 9(6)            PJJPLS
      VALUE                                     ZERO.      PJJPLS
      10      WR02-DORDE.                          PJJPLS
      11      WR02-DORDEJ PICTURE XX             PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-SLASH1 PICTURE X              PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-DORDEM PICTURE XX            PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-SLASH2 PICTURE X              PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WR02-DORDEA PICTURE XX            PJJPLS
      VALUE                                     SPACE.    PJJPLS
01      WK00.                                PJJPLS
      02      WK00-00.                              PJJPLS
      10      WK00-CLECD.                          PJJPLS
      11      WK00-COCARA PICTURE X              PJJPLS
      VALUE                                     SPACE.    PJJPLS
      11      WK00-NUCOM PICTURE 9(5)            PJJPLS
      VALUE                                     ZERO.      PJJPLS
      11      WK00-FOURNI PICTURE X(3)           PJJPLS
      VALUE                                     SPACE.    PJJPLS
      02      WK10.                                PJJPLS
      10      WK10-QTMAC PICTURE 99              PJJPLS
      VALUE                                     ZERO.      PJJPLS
      10      WK10-QTMAL PICTURE 99             PJJPLS
      VALUE                                     ZERO.      PJJPLS
      10      WK10-INFOR PICTURE X(35)          PJJPLS
      VALUE                                     SPACE.    PJJPLS
      10      WK10-RFI PICTURE X(100)           PJJPLS
      VALUE                                     SPACE.    PJJPLS
      02      WK20.                                PJJPLS
      10      WK20-EDIT PICTURE X               PJJPLS
      VALUE                                     SPACE.    PJJPLS
      02      WK30.                                PJJPLS
      10      WK30-NUCOM PICTURE 9(5)           PJJPLS
      VALUE                                     ZERO.      PJJPLS
01      DEBUT-WSS.                              PJJPLS
      05      FILLER PICTURE X(7) VALUE 'WORKING'. PJJPLS
      05      IK PICTURE X.                      PJJPLS
01      CONSTANTES-PAC.                        PJJPLS
      05      NUGNA PICTURE X(5) VALUE '0902 '. PJJPLS
      05      APPLI PICTURE X(3) VALUE 'MVF'.    PJJPLS
      05      DATGN PICTURE X(8) VALUE '12/04/91'. PJJPLS
      05      PROGR PICTURE X(6) VALUE 'PJJPLS'. PJJPLS
      05      CODUTI PICTURE X(6) VALUE 'PTJCQ '. PJJPLS
      05      TIMGN PICTURE X(8) VALUE '10:52:36'. PJJPLS

```

DESCRIPTION D'UN PROGRAMME GENERALE
DEBUT DE WORKING-STORAGE SECTION

PAGE

148

8
5

| | | | | | |
|----|--------|-------------|-------------|--|--------|
| 01 | DATCE. | | | | PJJPLS |
| 05 | CENTUR | PICTURE XX | VALUE '19'. | | PJJPLS |
| 05 | DATOR. | | | | PJJPLS |
| 10 | DATOA | PICTURE XX. | | | PJJPLS |
| 10 | DATOM | PICTURE XX. | | | PJJPLS |
| 10 | DATOJ | PICTURE XX. | | | PJJPLS |

8.6. VARIABLES ET INDICES

VARIABLES ET INDICES

INDICATEURS DE FIN DE FICHIER

XX-FT :

Fin de travail sur le fichier XX.

.Signifie que tous les enregistrements du fichier XX ont été traités y compris le dernier.

.Généré pour tout fichier traité séquentiellement d'utilisation C, M, N, P et pour tout fichier d'utilisation T ou X en organisation W ou L.

INDICES

Ils concernent le stockage et la recherche en table.

IXXNNM :

mémorise le nombre d'enregistrements maximum prévu par l'utilisateur (initialisé par VALUE).

IXXNNL :

mémorise le nombre d'enregistrements effectivement stockés dans la table issue de la description de l'enregistrement NN du fichier XX (initialisé à zéro, il doit être incrémenté au chargement).
Ce nombre ne peut excéder le nombre maximum.

IXXNNR :

variant de 1 à IXXNNL, est utilisé pour toute recherche sur la table issue de la description de l'enregistrement NN du fichier XX.

Ces trois indices sont générés pour tous les enregistrements des fichiers définis comme table, dans le cas de fichier à type de description étalée avec OCCURS pour les enregistrements ayant un nombre maximum d'enregistrements précisé, ou bien si on a défini une table W-XXNN dans les zones de travail de l'utilisateur ('-W').

COMPTEURS D'ENREGISTREMENTS

5-XX00-CPTENR :

Compteur d'enregistrements du fichier XX. Ce compteur est généré pour tout fichier d'utilisation autre que T ou X.
Il n'est pas incrémenté par les opérateurs de fichier et doit être géré explicitement.

CODES RETOUR DES FICHIERS

Pour chaque fichier en organisation V pour lequel on aura précisé dans la ligne d'appel de structure de données un FILE-STATUS on trouvera générée la zone 1-XXNN-CORUB où XXNN et CORUB sont les valeurs fournies dans la zone complémentaire. Le test du FILE-STATUS n'est pas effectué par les opérateurs de traitement de fichier.

DESCRIPTION D'UN PROGRAMME GENERE
VARIABLES ET INDICES

PAGE

151

8
6

| | | | | | |
|----|-------------------------------------|---------------|----------------------------|--|--------|
| 01 | VARIABLES-CONDITIONNELLES. | | | | PJJPLS |
| | 05 | FT. | | | PJJPLS |
| | 10 | LK-FT | PICTURE X VALUE '0'. | | PJJPLS |
| | 10 | TR-FT | PICTURE X VALUE '0'. | | PJJPLS |
| 01 | INDICES COMPUTATIONAL SYNC. | | | | PJJPLS |
| | 05 | ILB00L | PICTURE S9(4) VALUE ZERO. | | PJJPLS |
| | 05 | ILB00R | PICTURE S9(4) VALUE ZERO. | | PJJPLS |
| | 05 | ILB00M | PICTURE S9(4) VALUE +0012. | | PJJPLS |
| | 05 | ILB10L | PICTURE S9(4) VALUE ZERO. | | PJJPLS |
| | 05 | ILB10R | PICTURE S9(4) VALUE ZERO. | | PJJPLS |
| | 05 | ILB10M | PICTURE S9(4) VALUE +0100. | | PJJPLS |
| | 05 | ILB20L | PICTURE S9(4) VALUE ZERO. | | PJJPLS |
| | 05 | ILB20R | PICTURE S9(4) VALUE ZERO. | | PJJPLS |
| | 05 | ILB20M | PICTURE S9(4) VALUE +0100. | | PJJPLS |
| 01 | COMPTEURS-FICHIERS COMPUTATIONAL-3. | | | | PJJPLS |
| | 05 | 5-KL00-CPTENR | PICTURE S9(9) VALUE ZERO. | | PJJPLS |
| | 05 | 5-LA00-CPTENR | PICTURE S9(9) VALUE ZERO. | | PJJPLS |
| | 05 | 5-LK00-CPTENR | PICTURE S9(9) VALUE ZERO. | | PJJPLS |
| | 05 | 5-TR00-CPTENR | PICTURE S9(9) VALUE ZERO. | | PJJPLS |
| | 05 | 5-WK00-CPTENR | PICTURE S9(9) VALUE ZERO. | | PJJPLS |
| 01 | ZONES-STATUS. | | | | PJJPLS |
| | 05 | 1-LA00-STATUS | PICTURE XX VALUE ZERO. | | PJJPLS |
| | 05 | 1-TR00-STATUS | PICTURE XX VALUE ZERO. | | PJJPLS |

8.7. ZONES DE TRAVAIL ET DE LIEN

ZONES DE TRAVAIL ET DE LIEN

TABLES

A chaque fichier table en entrée est associée une zone en 'WORKING-STORAGE SECTION'.

A chaque enregistrement faisant l'objet d'une mise en table correspond une réservation sous la forme:

1-XXNN OCCURS MM, ou :

XX est le code fichier

NN est le code enregistrement

MM est le nombre de postes maximum prévu.

Si le fichier a été défini en utilisation T toutes les Rubriques apparaissent dans cette zone et seront stockées. Si le fichier a été défini en utilisation X seules les Rubriques différentes de FILLER apparaissent dans la description.

Chaque Rubrique apparaît sous la forme 1-XXNN-CORUB. Les Rubriques de la partie commune sont décrites dans chaque partie spécifique.

L'utilisateur peut modifier le contenu de ces zones de travail par des lignes d'appel de Structures de Données. En aucun cas il ne peut modifier leur emplacement dans le Programme.

ZONES DE TRAVAIL DE L'UTILISATEUR

Elles contiennent les zones ou noms de sections définies sur les lignes '-W' dont les deux premières positions du numéro de ligne ne sont pas alphabétiques. (Si elles sont alphabétiques, les lignes sont insérées en tout début de la WORKING-STORAGE).

Elles incluent aussi les descriptions des Structures de Données en organisation L avec emplacement non alphabétique ou appelées en lignes '-W'.

En jouant sur le nombre niveau et sur l'emplacement, l'utilisateur peut obtenir la description de la Structure de Données sous un niveau 01 ou dans une section (LINKAGE, COMMUNICATION, ...) qu'il aura lui-même introduite sur une ligne '-W'. Pour une structure de données d'organisation L ou appelée en ligne '-W', tous les types de descriptions sont possibles dans cette zone.

DESCRIPTION D'UN PROGRAMME GENERE
ZONES DE TRAVAIL ET DE LIEN

8
7

| | | | | |
|---|--|---------------------------|-------------|--------|
| 01 | | 1-LB-TABLE. | | PJJPLS |
| 02 | | 1-LB00T. | | PJJPLS |
| 05 | | 1-LB00 OCCURS | 0012. | PJJPLS |
| 10 | | 1-LB00-ANNUL PICTURE | X. | PJJPLS |
| 10 | | 1-LB00-MOIS PICTURE | 99. | PJJPLS |
| 10 | | 1-LB00-LMOIS PICTURE | X(9). | PJJPLS |
| 10 | | 1-LB00-FILLER PICTURE | X(68). | PJJPLS |
| 10 | | 1-LB00-ACTIO PICTURE | X. | PJJPLS |
| 10 | | 1-LB00-FLM PICTURE | 999. | PJJPLS |
| 10 | | 1-LB00-FLDDEP PICTURE | X(6). | PJJPLS |
| 10 | | 1-LB00-STRUCT PICTURE | X. | PJJPLS |
| 10 | | 1-LB00-PANFI PICTURE | X. | PJJPLS |
| 10 | | 1-LB00-PANL PICTURE | X(12). | PJJPLS |
| 02 | | 1-LB10T. | | PJJPLS |
| 05 | | 1-LB10 OCCURS | 0100. | PJJPLS |
| 10 | | 1-LB10-FLDCAN PICTURE | X(6). | PJJPLS |
| 10 | | 1-LB10-FLCADE PICTURE | X(3). | PJJPLS |
| 10 | | 1-LB10-FLCAAR PICTURE | X(3). | PJJPLS |
| 10 | | 1-LB10-FLTDEP PICTURE | X(6). | PJJPLS |
| 10 | | 1-LB10-FLTARR PICTURE | X(6). | PJJPLS |
| 10 | | 1-LB10-ACMSER PICTURE | 9(6). | PJJPLS |
| 10 | | 1-LB10-FLCFS PICTURE | X. | PJJPLS |
| 10 | | 1-LB10-FLQSTP PICTURE | 9. | PJJPLS |
| 10 | | 1-LB10-FILLER PICTURE | X(18). | PJJPLS |
| 02 | | 1-LB20T. | | PJJPLS |
| 05 | | 1-LB20 OCCURS | 0100. | PJJPLS |
| 10 | | 1-LB20-PANTTL PICTURE | XXX. | PJJPLS |
| 10 | | 1-LB20-PAMPHH PICTURE | X(10). | PJJPLS |
| 10 | | 1-LB20-PAMPHW PICTURE | X(10). | PJJPLS |
| 10 | | 1-LB20-RECCLA PICTURE | X. | PJJPLS |
| 10 | | 1-LB20-RECSMK PICTURE | X. | PJJPLS |
| 10 | | 1-LB20-REDCAN PICTURE | X(6). | PJJPLS |
| 10 | | 1-LB20-FILLER PICTURE | X(19). | PJJPLS |
| 01 | | ZONES-UTILISATEUR PICTURE | X. | PJJPLS |
| *SD: WG BIB: WG SEL: 99_____ FORM: I DESC: _ NIV: 1 ORG: _ SS: _ 720010 | | | | |
| 01 | | WG00. | | PJJPLS |
| 05 | | WG00-SUITE. | | PJJPLS |
| 15 | | FILLER PICTURE | X(00010). | PJJPLS |
| 01 | | WG99 REDEFINES | WG00. | PJJPLS |
| 10 | | WG99-MREEL9 PICTURE | S9(5)V99 | PJJPLS |
| | | COMPUTATIONAL-3. | | PJJPLS |
| 10 | | WG99-MREELX | | PJJPLS |
| | | REDEFINES | WG99-MREEL9 | PJJPLS |
| | | PICTURE | X(7). | PJJPLS |
| 10 | | WG99-DALI PICTURE | X(6). | PJJPLS |
| *SD: CC BIB: CD SEL: _____ FORM: I DESC: 4 NIV: 3 ORG: _ SS: _ 720020 | | | | |
| 01 | | CC00. | | PJJPLS |
| 02 | | CC00T. | | PJJPLS |
| 03 | | CC00-00. | | PJJPLS |
| 10 | | CC00-CLECD. | | PJJPLS |
| 11 | | CC00-COCARA PICTURE | X. | PJJPLS |
| 11 | | CC00-NUCOM PICTURE | 9(5). | PJJPLS |
| 11 | | CC00-FOURNI PICTURE | X(3). | PJJPLS |
| 02 | | CC05T. | | PJJPLS |
| 03 | | CC05. | | PJJPLS |
| 10 | | CC05-NUCLIE PICTURE | 9(8). | PJJPLS |
| 10 | | CC05-DATE PICTURE | X(6). | PJJPLS |
| 10 | | CC05-RELEA PICTURE | X(3). | PJJPLS |
| 10 | | CC05-REFCLI PICTURE | X(30). | PJJPLS |
| 10 | | CC05-RUE PICTURE | X(40). | PJJPLS |
| 10 | | CC05-COPOS PICTURE | X(5). | PJJPLS |
| 10 | | CC05-VILLE PICTURE | X(20). | PJJPLS |
| 10 | | CC05-CORRES PICTURE | X(25). | PJJPLS |
| 10 | | CC05-REMIS PICTURE | S9(4)V99. | PJJPLS |
| 10 | | CC05-MATE PICTURE | X(8). | PJJPLS |
| 10 | | CC05-LANGU PICTURE | X. | PJJPLS |
| 10 | | CC05-FILLER PICTURE | X(5). | PJJPLS |
| 02 | | CC10T. | | PJJPLS |
| 03 | | CC10. | | PJJPLS |
| 10 | | CC10-QTMAL PICTURE | 99. | PJJPLS |
| 10 | | CC10-QTMAL PICTURE | 99. | PJJPLS |
| 10 | | CC10-INFOR PICTURE | X(35). | PJJPLS |
| 10 | | CC10-RFI PICTURE | X(100). | PJJPLS |
| 02 | | CC20T. | | PJJPLS |
| 03 | | CC20. | | PJJPLS |
| 10 | | CC20-EDIT PICTURE | X. | PJJPLS |
| 02 | | CC30T. | | PJJPLS |
| 03 | | CC30. | | PJJPLS |

```

10          CC30-NUCOM PICTURE 9(5).          PJJPLS
*SD: KA BIB: CI SEL: _____ FORM: I DESC: _ NIV: 2 ORG: _ SS: _ 730010
02          KA00.          PJJPLS
10          KA00-CLECI.    PJJPLS
11          KA00-NOCL.    PJJPLS
12          KA00-NOCL11 PICTURE X.          PJJPLS
12          KA00-NOCL12 PICTURE XX.         PJJPLS
12          KA00-NOCL2 PICTURE XX.         PJJPLS
10          KA00-NOMCL PICTURE X(20).      PJJPLS
10          KA00-ADRES PICTURE X(43).      PJJPLS
10          KA00-NUDEP PICTURE XXX.        PJJPLS
10          KA00-LIDEP PICTURE X(24).      PJJPLS
10          KA00-NUREG PICTURE XXX.       PJJPLS
10          KA00-LIREG PICTURE X(24).      PJJPLS
PROCEDURE DIVISION.          PJJPLS
DECLARATIVES.                PJJPLS
SECLA SECTION.               PJJPLS
    USE AFTER ERROR PROCEDURE ON LA-FICHER. PJJPLS
FOALA. DISPLAY 'STATUS : ENT02 = ' 1-LA00-STATUS. PJJPLS
FOALA-A. GO TO F0A90.        PJJPLS
FOALA-FN. EXIT.              PJJPLS
SECTR SECTION.               PJJPLS
    USE AFTER ERROR PROCEDURE ON TR-FICHER. PJJPLS
FOATR. DISPLAY 'STATUS : ENT01 = ' 1-TR00-STATUS. PJJPLS
FOATR-A. GO TO F0A90.        PJJPLS
FOATR-FN. EXIT.              PJJPLS
FOA90. STOP 'INPUT-OUTPUT ERROR. CANCEL THE JOB'. PJJPLS
FOA90-FN. EXIT.              PJJPLS
END DECLARATIVES.           PJJPLS
SEC00 SECTION.               PJJPLS
N01. NOTE *****.          PJJPLS
      *                               *          PJJPLS
      *INITIALISATIONS                *          PJJPLS
      *                               *          PJJPLS
      *****.                      PJJPLS
F01. EXIT.                    PJJPLS
N01KL. NOTE *INITIALISATION FICHER KL-FICHER *. PJJPLS
F01KL. OPEN OUTPUT            KL-FICHER.    PJJPLS
F01KL-FN. EXIT.              PJJPLS
N01LA. NOTE *INITIALISATION FICHER LA-FICHER *. PJJPLS
F01LA. OPEN I-O              LA-FICHER.    PJJPLS
      IF 1-LA00-STATUS NOT = ZERO          PJJPLS
      PERFORM F0ALA            PJJPLS
      PERFORM F0A90           THRU F0A90-FN. PJJPLS
F01LA-FN. EXIT.              PJJPLS
N01LB. NOTE *INITIALISATION FICHER LB-FICHER *. PJJPLS
F01LB. OPEN INPUT            LB-FICHER.    PJJPLS
F01LB-10. READ LB-FICHER AT END          PJJPLS
      GO TO F01LB-20.          PJJPLS
      ADD 1 TO ILB00L          IF ILB00L NOT > 0012 PJJPLS
      MOVE LB00                PJJPLS
      TO 1-LB00                (ILB00L).      PJJPLS
      IF LB00-                  = 'F'        PJJPLS
      NEXT SENTENCE            ELSE GO TO F01LB-10F. PJJPLS
      ADD 1 TO ILB10L          IF ILB10L NOT > 0100 PJJPLS
      MOVE LB10                PJJPLS
      TO 1-LB10                (ILB10L).      PJJPLS
F01LB-10F.                    PJJPLS
      IF LB00-                  = 'P'        PJJPLS
      NEXT SENTENCE            ELSE GO TO F01LB-20F. PJJPLS
      ADD 1 TO ILB20L          IF ILB20L NOT > 0100 PJJPLS
      MOVE LB20                PJJPLS
      TO 1-LB20                (ILB20L).      PJJPLS
F01LB-20F.                    PJJPLS
      GO TO F01LB-10.          PJJPLS
F01LB-20.                      PJJPLS
      IF ILB00L > ILB00M          PJJPLS
      MOVE ILB00L TO ILB00R      PJJPLS
      MOVE ILB00M TO ILB00L.    PJJPLS
      IF ILB10L > ILB10M          PJJPLS
      MOVE ILB10L TO ILB10R      PJJPLS
      MOVE ILB10M TO ILB10L.    PJJPLS
      IF ILB20L > ILB20M          PJJPLS
      MOVE ILB20L TO ILB20R      PJJPLS
      MOVE ILB20M TO ILB20L.    PJJPLS
F01LB-99. CLOSE LB-FICHER.      PJJPLS
F01LB-FN. EXIT.              PJJPLS
N01LK. NOTE *INITIALISATION FICHER LK-FICHER *. PJJPLS

```

DESCRIPTION D'UN PROGRAMME GENERE
ZONES DE TRAVAIL ET DE LIEN

PAGE

155

8
7

| | | | |
|----------------|---------------------------------------|----------------|--------|
| F01LK-FN. | EXIT. | | PJJPLS |
| N01TR. | NOTE *INITIALISATION FICHIER | TR-FICHIER * | PJJPLS |
| F01TR. | OPEN INPUT | TR-FICHIER. | PJJPLS |
| | IF 1-TR00-STATUS | NOT = ZERO | PJJPLS |
| | PERFORM F0ATR | | PJJPLS |
| | PERFORM F0A90 | THRU F0A90-FN. | PJJPLS |
| F01TR-FN. | EXIT. | | PJJPLS |
| F01-FN. | EXIT. | | PJJPLS |
| * | NOTE * DEBUT ITERATION DU PROGRAMME | * | PJJPLS |
| F05. | EXIT. | | PJJPLS |
| N05. | NOTE ***** | | PJJPLS |
| | * | | PJJPLS |
| | *LECTURE FICHIERS ACCES SEQ. SANS DE* | | PJJPLS |
| | * | | PJJPLS |
| | ***** | | PJJPLS |
| N05LK. | NOTE *LECTURE FICHIER | LK SANS DE* | PJJPLS |
| F05LK. | IF LK-FT = '0' | | PJJPLS |
| | NEXT SENTENCE ELSE GO TO | F05LK-FN. | PJJPLS |
| F05LK-10. | RETURN LK-FICHIER | AT END | PJJPLS |
| | MOVE 1 TO | LK-FT | PJJPLS |
| | GO TO F05LK-FN. | | PJJPLS |
| | ADD 1 TO 5-LK00-CPTENR. | | PJJPLS |
| F05LK-FN. | EXIT. | | PJJPLS |
| N05TR. | NOTE *LECTURE FICHIER | TR SANS DE* | PJJPLS |
| F05TR. | IF TR-FT = '0' | | PJJPLS |
| | NEXT SENTENCE ELSE GO TO | F05TR-FN. | PJJPLS |
| F05TR-10. | READ TR-FICHIER | AT END | PJJPLS |
| | MOVE 1 TO | TR-FT | PJJPLS |
| | GO TO F05TR-FN. | | PJJPLS |
| | ADD 1 TO 5-TR00-CPTENR. | | PJJPLS |
| F05TR-FN. | EXIT. | | PJJPLS |
| F05-FN. | EXIT. | | PJJPLS |
| N20. | NOTE ***** | | PJJPLS |
| | * | | PJJPLS |
| | *FIN DE TRAITEMENT | | PJJPLS |
| | * | | PJJPLS |
| | ***** | | PJJPLS |
| F20. | IF FT = | ALL '1' | PJJPLS |
| | NEXT SENTENCE ELSE GO TO | F20-FN. | PJJPLS |
| F20KL. | CLOSE KL-FICHIER. | | PJJPLS |
| F20KL-FN. | EXIT. | | PJJPLS |
| F20LA. | CLOSE LA-FICHIER. | | PJJPLS |
| F20LA-FN. | EXIT. | | PJJPLS |
| F20TR. | CLOSE TR-FICHIER. | | PJJPLS |
| F20TR-FN. | EXIT. | | PJJPLS |
| F20-FN. | EXIT. | | PJJPLS |
| N90. | NOTE ***** | | PJJPLS |
| | * | | PJJPLS |
| | * ECRITURES | | PJJPLS |
| | * | | PJJPLS |
| | ***** | | PJJPLS |
| F90. | EXIT. | | PJJPLS |
| N90KL. | NOTE *ECRITURE SUR LE FICHIER KL | * | PJJPLS |
| F90KL. | | | PJJPLS |
| | MOVE LK00 | TO KL00. | PJJPLS |
| | WRITE KL05. | | PJJPLS |
| | WRITE KL10. | | PJJPLS |
| | WRITE KL20. | | PJJPLS |
| | WRITE KL30. | | PJJPLS |
| F90KL-99. | ADD 1 TO 5-KL00-CPTENR. | | PJJPLS |
| F90KL-FN. | EXIT. | | PJJPLS |
| F90-FN. | EXIT. | | PJJPLS |
| F9099-ITER-FN. | GO TO F05. | | PJJPLS |

9. ANNEXES

9.1. SOURCE COBOL (LIGNES '9')

LANGAGE SOURCE COBOL (LIGNES '9')

Les lignes '9' sont du langage source COBOL.

Elles permettent :

- . d'utiliser les descriptions de données générées par PACBASE dans des programmes écrits en COBOL,
- . d'utiliser les fonctions bibliothécaire et documentaire de PACBASE pour gérer les sources COBOL existants.

GENERATION DE DONNEES

Il est possible de faire générer par PACBASE les trois premières divisions COBOL d'un Programme et d'écrire la procédure en lignes '-9'.

Le Programme sera défini avec la nature de programme 'D' et la variante correspondant au compilateur visé.

Il sera constitué de lignes d'appel de Structures de Données, de lignes '-B' et '-W' et de lignes '-9' pour la procédure.

Il est également possible de ne faire générer que les descriptions de données et d'écrire le reste du programme en lignes '9'.

Le Programme sera défini avec la nature 'F' et la variante correspondant au compilateur visé.

Il sera constitué de lignes d'appel de Structures de Données et de lignes '-9'. L'emplacement des descriptions générées est déterminé, pour chaque ligne d'appel de Structure de Données, par une ligne '-9' comportant dans la colonne 7 du COBOL un 'F' suivi du code du fichier dans le programme (2 caractères). Les descriptions générées ne comportent pas la clause 'FD' et commencent au premier niveau 01.

SOURCE COBOL PUR

Il est possible de gérer dans la base PACBASE du langage source COBOL pur. Il en sera extrait en faisant 'générer' le programme avec la variante 'C'.

Un programme peut exister en deux versions: COBOL et PACBASE (lignes '-CD', '-CP', '-D', '-W', '-P' ou '-8'). Dans ce cas:

- . si la variante est C, les lignes '9' du programme seront extraites, les lignes de programmation PACBASE étant ignorées,
- . si la variante correspond à un COBOL généré, le programme sera généré, les lignes COBOL seront ignorées.

Si l'utilisateur veut voir apparaître dans son programme des renseignements identiques à ceux qu'il trouvera dans tout programme généré dans la zone CONSTANTES-PAC, il faut qu'il codifie une ligne 9 de la façon suivante :

```
01 CONSTANTES-PAC PIC X(20) VALUE '20 caractères'  
cols 12                               35           48
```

Par exemple :

```
01 CONSTANTES-PAC PIC X(20) VALUE  
'XXXXXXXXXXXXXXXXXXXXX'
```

Le résultat donné est de la forme AAA9999V000 DDDDDDDD, où

- . (AAA) est le code application,
- . (9999V) le numéro de la session au cours de laquelle le programme a été extrait,
- . (DDDDDDDD) la date de l'extraction.

Si l'utilisateur veut voir apparaître des informations plus complètes (code utilisateur, code de la base, date de génération avec siècle...), il doit coder une ligne 9 de la façon suivante :

```
cols 12                               35  
01 CONSTANTES-PAC PICTURE X(60) VALUE  
   'PACBASE-C20  
-   'DATGNC   ' .
```

Le résultat est composé des éléments suivants concaténés :

- . Numéro de session de génération (5 car.),
- . Code application (3 car.),
- . Date de la génération (8 car.),
- . Code Programme en Bibliothèque (6 car.),
- . Code utilisateur (8 car.),
- . Heure de génération (8 car.),
- . Nom externe du Programme (8 car.),
- . Code de la base (4 car.),
- . Date de la génération avec siècle (10 car.).

CODE OPERATION

C1: valeur par défaut.

C2: affichage de la source et d'une zone complémentaire saisissable.

```
-----  
!                                     !  
!           SUIVI PERMIS DE CONSTRUIRE           P0*DOC.LURE.JIA.893 !  
! TRAITEMENTS PROGRAMME   LTVAL1 FIRST TEST PROGRAM   !  
!                               1                               !  
!                                     !  
! 2 3      4 6      5                                     !  
! A LIGNE  S COMPL  DEBUT INSTRUCTION COBOL           : SOURCE BIBLI !  
! 006000 * PJJPS1  DUPLICATE RECORD VALIDATION       0318 !  
! 006001 PJJPS1  F36.                                0318 !  
! 006002 PJJPS1    IF CATX = 'HA'                     0318 !  
! 006003 PJJPS1    MOVE '1' TO W-WW00-REC             0318 !  
! 006004 PJJPS1    ELSE MOVE ZERO TO W-WW00-REC.     0318 !  
! 006005 PJJPS1  F36-FN.                              0318 !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
!                                     !  
! *** FIN ***                                         !  
! O: C2 CH: -9                                         !  
-----
```

| NUM | LON | CLASSE VALEUR | SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE |
|-----|-----|------------------|--|
| 1 | 6 | | CODE PROGRAMME Code identifiant le programme dans la bibliothèque. |
| 2 | 1 | | CODE ACTION |
| 3 | 6 | | NUMERO DE LIGNE COBOL ZONE PSEUDO-NUMERIQUE Les lignes peuvent être numérotées à l'introduction, ou renumérotées après mise à jour, avec un pas de 100 en COBOL. Dans une macro-structure, le numéro de ligne COBOL peut être paramétré sur les 4 chiffres de gauche par tranche de 2 chiffres. |
| 4 | 1 | - * | SUITE (COLONNE 7 DU COBOL) Suite de littéral (utilisation normale du COBOL). Commentaire (COBOL ANS seulement). |
| 5 | 65 | | INSTRUCTION COBOL Première partie de la ligne COBOL. La seconde partie (Colonnes 66 à 72) peut être visualisée avec le Code Opération C2. |
| 6 | 6 | | FIN DE LA LIGNE COBOL Correspond à la zone d'identification du programme du format COBOL. |

SAISIE EN BATCH

Le source COBOL est décrit à l'aide du bordereau 9.

Cette ligne doit toujours être précédée d'une ligne de définition de Programme car elle ne comporte pas le code Programme.

C = création de la ligne dans la bibliothèque.

M = modification de la ligne.

Blanc = création ou modification,
suivant état de la bibliothèque.

A = annulation de la ligne.

B = annulation de plusieurs lignes dans le Programme à
partir du numéro de ligne indiqué inclus.

R = fin d'annulation multiple au-delà de cette ligne.