

VisualAge Pacbase



Module Langage Structuré

Version 3.5



VisualAge Pacbase



Module Langage Structuré

Version 3.5

Note

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Notices», à la page v.

Vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir de :

<http://www.ibm.com/support/docview.wss?rs=37&context=SSEP67&uid=swg27005478>

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

Deuxième édition (Novembre 2006)

La présente édition s'applique à :

- VisualAge Pacbase Version 3.5

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.ibm.com/software/awdtools/vapacbase/support.html> ou en nous adressant un courrier à :

IBM Paris Laboratory
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, 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 1983,2006. All rights reserved.

Table des matières

Notices	v	Description des traitements (-P)	59
Marques	vii	Chapitre 6. Titres, conditions et niveaux des traitements	89
Chapitre 1. Introduction	1	Titres et conditions des traitements (-TC)	89
Objet du manuel	1	Titres et niveaux des traitements (-TO)	100
Principes de Description	1	Chapitre 7. Répertoire d'accès	105
Langage Structuré : Présentation	2	Accès aux Ecrans	105
Entité et lignes gérées	3	Options d'affichage en TP	109
Chapitre 2. Programmes	5	Codes action en TP	110
Programmes : Présentation	5	Edition et génération	111
Structure d'un Programme	5	Chapitre 8. Description d'un Programme généré	113
Définition d'un Programme	7	Introduction	113
Chapitre 3. Macro-Structures	15	Identification Division	113
Macro-Structures : Présentation	15	Environment Division	114
Appel de Macro-Structures (-CP)	18	Data Division File Section	115
Macro-Structures non ventilées	23	Début de Working Storage Section	118
Chapitre 4. Emploi des Structures de Données	25	Variables et indices	121
Appel de Structures de Données (-CD)	25	Zones de travail et de lien	122
Modification du début de Programme (-B)	47	Chapitre 9. Annexes	129
Zones de travail et de lien (-W)	50	Source Cobol (Lignes '9')	129
Chapitre 5. Langage Structuré	59		

Notices

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 : IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504-1785, U.S.A.

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, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex, 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, VisualAge Pacbase, RACF, RS/6000, SQL/DS 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.

Chapitre 1. Introduction

Objet du manuel

Ce manuel décrit toutes les spécifications du module Langage Structuré utilisables sans les modules complémentaires Pacbench C/S, 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

Vous devez avoir suivi une session de formation aux modules Dictionnaire et Langage Structuré.

Vous devez également avoir pris connaissance :

- du manuel "Dictionnaire de Données",
- du guide de "l'Interface Utilisateur Mode Caractère".

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.

Pour la description de ces entrées en batch, reportez-vous au manuel des "Procédures du Développeur".

Dans la description, chaque zone est repérée par un numéro d'ordre correspondant à l'ordre de tabulation sur l'écran.

NOTE : Si vous utilisez la Station de Travail VisualAge Pacbase, il est recommandé de consulter le guide de "l'Interface Utilisateur Station de Travail" dans lequel sont documentées les fenêtres de la Station.

Langage Structuré : Présentation

Le Langage Structuré 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, ...),
- Une portabilité améliorée, par une conception indépendante de tout constructeur et une génération de Programmes COBOL adaptée à chaque type de matériel.

INTRODUCTION A L'ECRITURE DE CODE SPECIFIQUE (LIGNE -P)

Les traitements générés en standard ne peuvent répondre à tous les besoins applicatifs. Ils proposent des solutions standard à des problèmes standard.

Vous pouvez être amené à les compléter, parfois à les remplacer partiellement. Au préalable, il faut néanmoins vous demander si le traitement à effectuer ne correspond pas à un cas standard. A l'inverse, il est déconseillé de réduire à un traitement automatique, un cas dont la complexité nuirait à la clarté de la structure du Programme.

Lors de la réalisation d'un Programme, il est donc nécessaire de bien réfléchir à la nature des traitements requis et à leur spécificité. Pour cela, une bonne connaissance de la structure de la procédure du Programme généré est indispensable.

VisualAge Pacbase fournit un support pour l'écriture des traitements spécifiques : la ligne -P. Son utilisation permet d'assurer la structuration des traitements, la lisibilité des Programmes et leur portabilité.

DESCRIPTION DU MODULE

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 Structuré 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 Structuré.

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 Structuré 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 Structuré.

Entité et lignes gérées

Le module Langage Structuré 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 ('-P') permet de définir son code, son nom et ses caractéristiques principales.

Les lignes d'appel de Macro-Structures ('-CP') 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 ('-CD') permettent d'obtenir la description des zones d'entrée-sortie du Programme.

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

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

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

NOTE : Pour les lignes d'accès batch, reportez-vous au manuel "Les Procédures du Développeur".

Comme toutes les entités VisualAge Pacbase, les Programmes peuvent être documentés par des lignes de Commentaires et par la ventilation de Textes (Cf. manuel "Dictionnaire de Données").

Chapitre 2. Programmes

Programmes : Présentation

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, mots-clés, variante de COBOL à générer, etc.).
- des lignes de commentaires, 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 (et aussi de la PROCEDURE 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 : le Dictionnaire,
- avec le module Langage Batch, l'obtention de fonctions automatiques (contrôle, mise à jour, édition) dans les Programmes générés.

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 VisualAge 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 VA PAC

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

- le numéro de session de génération du Programme,
- le code de la Bibliothèque où est défini le 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.
- la version du générateur,
- la date du générateur,
- la date du squelette.

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.

Définition d'un Programme

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

Il s'obtient par le choix :

CH: P.....

CODE PROGRAMME 1IPED1

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 AVEC '-' :13

MOTS CLES ASSOCIES.....:14 STATISTIQUES DOSSIER

MIS A JOUR PAR.....: LE : A : : : BIB :

NO DE SESSION.....: 0175 BIBLIOTHEQUE : JIA BLOCAGE :

0: C1 CH: Pjiped1

ACTION:

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	6		CODE PROGRAMME (OBLIGATOIRE)
			Code identifiant le Programme dans la Bibliothèque.
2	30		NOM 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 VARIANTE DU LANGAGE A GENERER.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			La compatibilité du généré avec Cobol 85, Cobol II, Cobol/370, Cobol OS/390 est assurée en fonction de la valeur de LANGAGE GENERE au niveau de la Bibliothèque.
		'N'	Pas d'adaptation à une variante de langage. Permet d'empêcher toute génération.
		'0'	IBM MVS/ESA OS/390
		'1'	IBM DOS/VSE
		'3'	UNIX WINDOWS
		'4'	BULL GCOS7
		'5'	BULL GCOS8
		'8'	UNISYS Serie A
		'F'	TANDEM
		'I'	DEC/VAX VMS
		'K'	ICL 2900
		'O'	AS/400
		'U'	UNISYS Serie 2200
		'X'	IBM MVS/ESA OS/390
		'Q'	ACUCOBOL
		'C'	Extraction de source COBOL (voir le manuel "Langage Structuré", chapitre "Annexes", sous-chapitre "Source COBOL")
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 instructions (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	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'5'	Suppression de la numérotation et de l'identification du Programme et cadrage des instructions (colonnes 8 à 71 incluses) en colonne 1.
6	1		Option cartes avant programme
			Code option du jeu de lignes de contrôle à insérer avant chaque Programme généré ; au niveau de la Bibliothèque, cette option représente les valeurs par défaut à affecter aux Programmes lors de leur création.
7	1		Option cartes après programme
			Code option du jeu de lignes de contrôle à insérer après chaque Programme généré ; au niveau de la Bibliothèque, cette option représente les valeurs par défaut à affecter aux Programmes lors de leur création.
8	8		CODE PROGRAMME GENERE
			(Valeur par défaut au niveau de la génération : code classement du Programme généré).
			Identifie le Programme généré : . Dans l'identification division. . En bibliothèque symbolique. . En bibliothèque de modules exécutables.
			Ce code intervient totalement ou partiellement dans les lignes de langage de contrôle générées avant ou après Programme.
9	1		TYPE DE PROGRAMMATION
			Langage Structuré
		'P'	Programmation en Langage Structuré par ligne de type '-P'. (Valeur par défaut lors de la création d'une Bibliothèque).
			Générateur COBOL (générateur Reverse)
		'S'	Procédure composée de lignes de Langage Source ('-SC') et de Traitements Structurés ('-P').
			La nature du Programme doit également être 'S'.
		'8'	Programmation par ligne de type '-8'.
			Réservé exclusivement à la maintenance d'applications écrites à l'aide de versions antérieures à la version VisualAge Pacbase.
			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.
			Au niveau d'un Programme, le type de programmation ne peut être modifié.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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.
10	1	NATURE DU PROGRAMME
		Elle définit la structure du Programme généré ou la nature du Programme en Bibliothèque.
	'B'	Structure d'un Programme 'batch' (Option par défaut).
		C'est la structure normale d'un Programme itératif :
		. Début de l'itération (F05)
		. Fin de travail (F20)
		. Fin d'itération (F9099. GO TO F05).
	'S'	Pas de structure de Programme imposée :
		Module Langage structuré :
		Cette nature peut être employée pour décrire un 'SYSTEME GENERATION' en TDS, un 'SCHEMA' en IDS II,...
		. 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'.
		Module Générateur COBOL :
		Cette nature est obligatoire pour un Programme de type 'S'.
		Le Programme est constitué à partir de lignes '-W', '-P', '-SC', et '-CP'.
	'T'	Structure d'un Programme transactionnel.
		Suppression de la notion d'itération c'est-à-dire :
		. Pas de génération de début de l'itération (F05).
		. Pas de fin de travail (F20).
		. Pas de fin d'itération (F9099. GO TO F05).
	'C'	Structure d'un Programme transactionnel sous C.I.C.S macro-level.
		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.)
		En outre :

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		.Génération en tête de procédure division de l'instruction : MOVE CSACDTA TO TCACBAR.
		.Génération en F9099 de : DFHPC TYPE=RETURN.
		.Pas de numérotation des lignes du programme généré.
	'M'	Macro-Structure Paramétrée.
		(Valeur documentaire uniquement).
		Un Programme de ce type est destiné à être incorporé dans un autre Programme et ne peut donc être généré seul.
	'F'	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 "Source COBOL" dans le manuel "Langage Structuré").
	'D'	Programme constitué de lignes '-CD', '-B', '-W' et '-9'.
		Cette nature permet d'obtenir la génération automatique de l'ensemble des divisions IDENTIFICATION ENVIRONMENT et DATA, la PROCEDURE DIVISION étant entièrement écrite par lignes '-9'.
	'P'	Programme constitué de lignes '-CD', '-B', '-W' et 'P' ou '-8' pour lequel seules les divisions IDENTIFICATION, ENVIRONMENT et DATA seront générées automatiquement, la PROCEDURE DIVISION étant entièrement obtenue à l'aide des lignes '-P' ou '-8'.
	'Y'	Programme écrit en langage C et constitué de lignes '-W', '-SC' et '-CP'.
11	1	TYPE DE L'ENTITE PROGRAMME
		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.
	'A'	Système TP.
	'M'	Macro-Structure.
	'N'	Macro-Structure non ventilée.
	'G'	Grille.
	'P'	Programme.
	'S'	Schéma.
	'T'	Transaction.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'V'	Sous-schéma.
		'U'	Utilitaire.
		'D'	Sous-programme.
12	1		CONTROLE DE PRESENCE ZONE NUMERIQUE
			Dans les Programmes de contrôle, la détermination de la présence des Rubriques numériques se fera suivant la valeur de ce code.
		BLANC	Présence de la Rubrique si elle n'est pas à blanc. (Valeur par défaut).
		'0'	Présence de la Rubrique si elle n'est pas à zéro.
		'L'	Présence de la Rubrique si elle n'est pas à low-value. Cette option concerne alors toutes les Rubriques.
13	1		GENERATION INDICATEURS SQL AVEC '-'
			Possibilité d'obtenir les chaînages sur l'utilisation des indicateurs SQL dans le Langage Structuré.
		BLANC	Génération des indicateurs SQL sous la forme : VXXNNCORUB.
		'-'	Génération des indicateurs SQL sous la forme : V-XXNN-CORUB.
14	55		MOTS CLES ASSOCIES
			Cette zone permet la saisie de mots-clés explicites, le système créant automatiquement des mots-clés implicites à partir du nom des entités.
			Les différents mots-clés doivent être séparés par au moins un blanc.
			La longueur maximum d'un mot-clé est de 13 caractères. Ceux-ci doivent être alphanumériques. Les caractères = et * sont ignorés.
			Majuscules et minuscules sont équivalentes, mais attention aux minuscules accentuées (e et E sont équivalents, mais pas é et E).
			NOTE : Les caractères accentués et les caractères spéciaux peuvent toutefois être déclarés équivalents à une valeur interne pour faciliter la recherche des instances par mots-clés (Administrator workbench, menu "Fenêtre", choix "Browser des Paramètres", onglet "Caractères spéciaux").

NUMLON		CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Il est possible d'affecter dix mots-clés explicites au maximum par entité. Pour plus de détails, voir le chapitre "Recherche d'instances", dans la partie dédiée aux mots-clés dans le "Guide de l'Interface Mode Caractère".

Chapitre 3. Macro-Structures

Macro-Structures : Présentation

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

Vous êtes 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 Commentaires (-GC) 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 essayera d'étudier les M.S.P. déjà existantes afin de minimiser les différences.

REMARQUE :

Tout Programme peut être utilisé comme Macro-Structure non paramétrée.

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.

INSTANCES D'ENTITES UTILISEES COMME PARAMETRES

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

Lorsqu'on appelle une de ces instances par l'intermédiaire des paramètres, mètres, aucun chaînage n'est établi si le nom de l'instance 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. 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.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Une Macro-Structure est un ensemble quelconque de lignes '-B', '-W', '-P' ou '-SC' (si elle est issue de Reverse Engineering).
			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.
			Si des lignes de deux Macro-Structures ont le même indicatif, elles sont toutes ignorées à la génération du Programme appelant.
			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.
			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.
			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.
4	2		NUMERO DE LIGNE
			ZONE NUMERIQUE PURE (Blancs et zéros équivalents).
			Il permet de définir plusieurs lignes de commentaire pour une Macro-Structure, ou encore, lorsque la Macro- Structure est paramétrée, de l'appeler plusieurs fois dans le même Programme, avec des valeurs différentes pour les paramètres.
5	1		SUITE
		*	Création d'une ligne suite.
6	50		COMMENTAIRES, VALEURS DES PARAMETRES
			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.
			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.

NUM	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LONG	VALEUR	REPLISSAGE
		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.
		Deux '/' consécutifs permettent d'ignorer le paramètre correspondant.
		Il est possible de créer une ligne suite par appel si les valeurs ne tiennent pas sur la première ligne.
		Le '/' est pris par défaut comme délimiteur mais il peut être remplacé par un caractère quelconque indiqué dans la zone DELIMITEUR DES VALEURS DE PARAMETRES.
		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).
		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.
7	1	DELIMITEUR DES VALEURS DE PARAMETRES
		C'est le caractère servant à séparer les différentes valeurs des paramètres. S'il n'est pas indiqué, le caractère '/' est pris par défaut.

Macro-Structures non ventilées

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

Chapitre 4. Emploi des Structures de Données

Appel de Structures de Données (-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

P0*AG.LILI.IBM.847

STRUCTURES DE DONNEES DU PROGRAMME 1 F020MA 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 EM
ED       : ED ED   SSFOU  0 R      I          1          I      1
        :   Z.COMPL.: 28          CLE ACC.: 29          IDENT. : 30
AC       : AC FORACH VSFID  0          T          I          1
        :   Z.COMPL.: AC00STATU CLE ACC.: NUACH          IDENT. :
CA       : AC SEQACH SSFOU  0          D          I          1
        :   Z.COMPL.:          CLE ACC.:          IDENT. :
FB       : FB FORFOU VRFID  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          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          I          1
        :   Z.COMPL.: XS00STATU CLE ACC.: XCLER7          IDENT. :
        :
        :   Z.COMPL.:          CLE ACC.:          IDENT. :
    
```

0: C1 CH: -CD

NUMLON	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
	'C'	Création de la ligne
	'M'	Modification de la ligne
	'A' ou 'D'	Annulation de la ligne
	'T'	Transfert de la ligne
	'B'	Annulation multiple
	'G'	Transfert d'un groupe de lignes
	'?'	Demande de documentation
	'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
	'X'	Mise à jour implicite sans transformation majuscule/ minuscule

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
3	2	CODE STRUCTURE DONNEES PROGRAMME (OBLIGATOIRE)
		Ce code établit la séquence de traitement de la S.D. dans le Programme.
		Le premier caractère doit être alphabétique mais le second peut être numérique ou alphabétique.
		Il est conseillé, lorsqu'une Structure de Données 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. dans le Programme.
		POUR L'ORGANISATION 'Y'
		Cette zone doit contenir le code de la clause COPY COBOL qui représente la zone de communication du Composant Applicatif Pacbench C/S accédant à la Vue Logique.
		Pour plus d'informations, référez-vous au manuel "Applications Pacbench C/S - Services Applicatifs".
		POUR LES ORGANISATIONS 'SQL'
		Cette zone doit contenir le code VisualAge Pacbase du bloc SQL.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Pour plus d'informations, référez-vous au manuel "Langage Structuré", chapitre "Langage Structuré", sous-chapitre "Description des Traitements (-P)" et au manuel "Bases de Données SQL", chapitre "Accès SQL", sous-chapitre "Personnalisation des accès SQL".
		POUR LES AUTRES ORGANISATIONS
		IBM OS/390 (variante X) :
		DDNAME sur 1 à 6 positions.
		IBM DOS (variante 1) trois formes :
		. 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.
		COBOL II IBM VSE (variante X) :
		La clause ASSIGN (pour les fichiers séquentiels, organisation 'S') de nom externe SYSnnn est générée sous forme : SYSnnn-UT-....-S-SYSnnn
		BULL Gcos7 (variante 4) :
		INTERNAL-FILE-NAME sur 1 à 6 positions.
		BULL Gcos8 (variante 5) :
		File-code (2 car.).
		UNISYS série A (variante 8) :
		NNPPP numérique, fait générer AREA nn AREASIZE pppp.
		TANDEM (variante F) :
		Nom externe sur 1 à 6 positions.
		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
	'V'	VSAM (IBM), UFAS (BULL), etc.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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 à votre choix.
		Voir la zone EMBLACEMENT 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 Pactables.
	'Y'	Appel de la Clause COPY qui correspond à la zone de communication entre le client et le serveur (Composants Applicatifs Pacbench C/S uniquement).
		Pour plus d'informations, référez-vous au Manuel "Applications Pacbench C/S - Services Applicatifs".
		BASES DE DONNEES
		Les valeurs de code suivantes sont réservées à la Description des Bases de Données pour les utilisateurs ne 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 II, (suivant la variante du Programme), dans la génération de DBD, SYSGEN, schémas ou Programmes d'application (suivant la nature du Programme).

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'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.
	'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, SYBASE et SQL SERVER.
		L'utilisation du Système avec les SGBD est développée dans des manuels spécifiques, auxquels vous vous reporterez pour de plus amples informations.
8	1	Mode d'accès
	'S'	Séquentiel (option par défaut).
	'R'	Direct.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'D'	Dynamique (Réservé 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
		'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,

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		. IBM MVS, la clause BLOCK CONTAINS est générée pour un fichier VSAM seulement si la Bibliothèque est en COBOL II.
		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)
		. Block 0 pour Unisys série A (variante 8), AS 400 (variante O).
		. Block 0 pour COBOL II IBM VSE et type de blocage à 'N'.
13	1	TYPE DE BLOCAGE
	'R'	Records (valeur par défaut).
	'C'	Characters.
	'N'	La clause BLOCK CONTAINS n'est pas générée.
14	1	NOMBRE DE NIVEAUX DE RUPTURE
		(Module Langage Batch)
	0 à 9	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).
		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.
		Les ruptures sur les niveaux majeurs seront automatiquement décelées. Le tout sera disponible sous forme de deux jeux de variables :
		. dd-PEn : premier Segment sur la S.D. dd au niveau n (valeur 1 oui, 0 non),
		. dd-DEn : dernier Segment sur la S.D. dd au niveau n (valeur 1 oui, 0 non),
		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) :

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			. RTPn pour la rupture entre l'itération précédente du Programme et celle en cours,
			. RTDn pour la rupture entre l'itération en cours du Programme et la prochaine,
			Ces variables sont également disponibles lorsqu'il n'y a pas de S.D. synchronisées dans le Programme.
			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.
15	1		NOMBRE DE NIVEAUX DE SYNCHRONISATION
		0 A 9	Les Structures de Données 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 contrôlé).
			Lorsqu'il y a plus de deux S.D., certaines peuvent être synchronisées à un niveau majeur par rapport aux autres.
			Valeur par défaut 0.
			Pour que la synchronisation soit automatique, les conditions suivantes doivent être réalisées :
			. 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).
			Si ces conditions ne sont pas réalisées, il faut modifier la cinématique du Programme par ligne -P.
			La synchronisation se traduit par deux jeux de variables donnant l'état des S.D. :

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			. 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 Segments de la S.D. dd.
16	1		UTILISATION DE LA S.D.
			Code permettant de définir le rôle de la S.D. dans le Programme, et commandant la génération des fonctions automatiques du module Langage Batch.
		'C'	Consultée
			S.D. quelconque en entrée.
		'D'	Divers
			S.D. quelconque en sortie (valeur par défaut).
		'T'	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 sélectionner 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 Langage Batch :
		'P'	Principale

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			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
		'M'	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 où 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.
		'T'	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 spool. Suppose l'utilisation de la Structure 00 pour définir une partie commune à toutes les lignes.
17	2		CORRESPONDANCE PRINCIPAL - RESULTAT
			Pour une S.D. principale (utilisation 'P'):
			indiquer le code dans le Programme de la S.D. résultat en sortie.
			Pour une S.D. résultat (utilisation 'R' ou 'D'):
			indiquer le code dans le Programme de la S.D. principale en entrée.
18	2		CORRESPONDANCE SOURCE - SELECTIONNE

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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).
		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').
		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.
19	1	NIVEAU DE RUPTURE SUR MOUVEMENT
		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 de la S.D. mouvement).
		Tant qu'aucune rupture n'est détectée à ce niveau, les enregistrements traités appartiennent au même mouvement.
		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).
		Par défaut, il est égal au nombre de niveaux de rupture de la S.D.
20	4	TYPE D'UNITE PHYSIQUE
		Généré dans la clause SELECT pour certains COBOLS :
		IBM VSE, 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
		ACU COBOL (variante Q) :
		.LS = génération de la clause LINE SEQUENTIAL
		BULL Gcos7 (variante 4) :
		.SSF = option WITH SSF dans la clause SELECT

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			.OUT = option suffixe -SYSOUT après le nom du fichier dans la clause SELECT (WITH SSF est généré).
			BULL Gcos8 (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.
			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 :
			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 2200 (variante U) :

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		.CR = lecteur de cartes
		.CP = perforateur de cartes
		.UN = UNISERVO
		.TP = bande
		.PN = imprimante avec nom externe. Si le champ 'complément du type d'unité physique' est renseigné, la clause RECORDING est également générée.
		.PT = imprimante sans nom externe
		.PF = imprimante avec nom externe et :
		VALUE OF PRINTER-FORMS 3-FF00-FORMS
		LINAGE IS 3-FF00-LINES
		TOP IS 3-FF00-TOP
		BOTTOM IS 3-FF00-BOTTOM
		Ces quatre noms-données sont à déclarer dans les zones de travail (-W) avec les valeurs appropriés.
		AS 400 (variante O) :
		.RD = lecteur
		.CP = perforateur de cartes
		.DB = base de données
		.PT = imprimante
		.TP = bande
		.DK ou blanc = disque
21	1	COMPLEMENT AU TYPE D'UNITE PHYSIQUE
		IBM DOS (variante 1) :
		.R = lecteur
		.P = perforateur
		BULL Gcos8 (variante 5) :
		.S = CODE SET EBCDIC
		.C = CODE SET ASCII
		UNISYS 2200 (variante U) :
		.S = RECORDING suivi du mode de blocage
		BULL Gcos7 (variante 4) et Gcos8 (variante 6) :

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			.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.
			DEC VAX VMS (variante I) :
			.A = Ouverture de fichier avec l'option ALLOWING ALL et lecture séquentielle avec l'option REGARDLESS.
			IBM MVS :
			.F = génération du paramètre OPTIONAL dans la clause SELECT d'un fichier VSAM.
22	9		ARGUMENT, SELECTION ETATS ENR.
			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.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Il est préférable, pour créer une S.D. composite, de sélectionner un Segment unique qui regroupera les Segments choisis par appel de Segment .
			Pour plus d'informations, voir le chapitre dédié aux Segments dans le manuel "Dictionnaire de Données".
			3. Sélection d'Etats :
			Pour sélectionner un Etat particulier, le troisième caractère du code de l'Etat à éditer dans le Programme doit être saisi dans la zone.
			Pour sélectionner tous les Etats ayant le même préfixe, la zone doit être laissée à blanc.
			Pour sélectionner plus de 9 Etats, créer une ligne 'suite' (voir la zone SUITE).
			Modification des codes de Segments et d'Etats :
			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 :
			1=2 changement du code Etat 1 en code Etat 2.
			*01=02 changement du code Segment 01 en code Segment 02.
23	1		Choix du type de format
			Le choix du type de format est réservé aux Structures de Données d'utilisation autre que 'I' ou 'J'.
		'E'	Format d'entrée
			Option par défaut pour les fichiers d'utilisation 'M', 'N', 'E'.
		'I'	Format interne
			Option par défaut pour les fichiers d'utilisation 'M', 'N', 'E'.
		'S'	Format d'édition
			Remarque : les Rubriques constituant les Segments ne doivent pas dépasser 999 caractères.
24	1		DESCRIPTION SELECTIONNEE
			Précise si les Rubriques facultatives (ENPR, GRPR, ERUT) figurant dans la description de la S.D. sont à reprendre :
		BLANC	Pas de prise en compte des Rubriques facultatives.
		'V'	Prise en compte de toutes les Rubriques facultatives.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'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).
		'E'	Prise en compte des Rubriques ENPR et GRPR seules.
		'U'	Prise en compte de la Rubrique ERUT seule.
			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.
			. 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.)
			La position supplémentaire est l'erreur d'identification. Elle initialise le vecteur EN-PRR.
			. GRPR (1 position par Segment + 1 pour l'erreur de groupe), initialise le vecteur GR-PRR.
			Dans une S.D. d'utilisation différente, elles peuvent figurer n'importe où et sont de format libre.
		'1..9,0'	Pour le module Pactables, indique le numéro de sous-schéma désiré.
			Pour un fichier d'utilisation SQL, indique le numéro de sous-schéma désiré (sélection d'une colonne d'une Table).
25	1		TYPE DE DESCRIPTION GENEREE
			Cette option permet d'obtenir plusieurs modes d'agencement des enregistrements d'une S.D. à partir des mêmes descriptions en Bibliothèque.
		BLANC	Enregistrements redéfinis (option par défaut).
		'1'	Enregistrements à la suite (partie commune suivie des différentes parties spécifiques), sans valeur initiale ni répétition d'enregistrements.
			Si la description de la S.D. apparaît en FILE SECTION COBOL, la zone NOMBRE NIVEAU DU SEGMENT doit avoir la valeur 2.
		'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.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Les valeurs initiales seront aussi générées pour les zones occursées si le "Langage généré" de la Bibliothèque est "D" (COBOL II, 85, ou 370).
		'3'	Enregistrements à la suite avec prise en compte du nombre de répétitions spécifié dans la définition des Segments, sans valeurs initiales.
			Si la Description de la S.D. apparaît en FILE SECTION COBOL, la Rubrique NOMBRE NIVEAU DU SEGMENT doit avoir la valeur 2.
			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').
		'4'	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.
			Remarque spécifique au module Dialogue :
			En cas de type de description '4' et de nombre niveau '3', l'indice n'est pas généré.
			Le niveau 2 permet d'accéder à la table constituée par la répétition d'un même enregistrement (FFEET).
			Le niveau 1 regroupe l'ensemble des informations de la S.D. (partie commune et parties spécifiques répétées ou non).
			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').
26	1		NOMBRE NIVEAU DU SEGMENT
			Cette option, employée en relation avec la précédente, définit le niveau COBOL des descriptions de S.D., de Segments ou de Rubriques.
			Dans les descriptions ci-après on entend par niveau S.D., le niveau COBOL de la zone DD00 (éventuellement 1-DD00, 2-DD00).
		1	Niveau 01 pour S.D. et Segments (valeur par défaut).
			Si la description de la S.D. apparaît en FILE SECTION COBOL, les Segments doivent obligatoirement être redéfinis.
			Si la Structure de Données est sans partie commune avec une description étalée, le niveau S.D. n'apparaît pas, sauf quand le type de description est à blanc.

NUM	MLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		2	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'.
		3	Niveau 02 pour S.D. et 03 pour Segments, lorsqu'il est associé aux types de description 1, 2 ou 3,
			Niveau 01 pour S.D. et 03 pour Segments lorsqu'il est associé au type de description 4,
			Niveau 03 pour S.D. et Segments lorsqu'il est associé au type de description blanc.
			Réservé aux S.D. en organisation 'W' ou 'L'.
		4	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.
			Réservé aux S.D. d'organisation 'L' et utilisation 'D'. Le niveau 01 est à définir par ligne '-W'.
		5	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.
			Réservé aux S.D. en organisation 'L' ou 'W' et utilisation 'D'.
		6	Niveau 02 pour les Rubriques de premier niveau.
			Les Rubriques élémentaires appartenant à une Rubrique groupe, les niveaux S.D et Segments disparaissent.
			Réservé aux S.D. d'organisation 'L' et utilisation 'D'
			Le niveau 01 est à définir par ligne '-W'.
			Valable uniquement pour un Ecran standard Dialogue.
		7	Niveau 01 pour les Rubriques de premier niveau.
			Les Rubriques élémentaires appartenant à une Rubrique groupe, les niveaux S.D et Segments disparaissent.
			Réservé aux S.D. en organisation 'L' ou 'W' et utilisation 'D'.
			Valable uniquement pour un Ecran standard Dialogue.
27	2		EMPLACEMENT DE LA DESCRIPTION

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			ZONE PSEUDO-NUMERIQUE, blancs et zéros équivalents.
			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).
			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 vous devez définir par ligne '-W'), ou en tête de la WORKING-STORAGE SECTION.
			Cette option est réservée aux S.D. en organisation 'L', 'D' ou 'W' et doit être généralement combinée avec les deux précédentes.
			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 "Langage Structuré").
		00	La description de la S.D. est insérée après toutes les lignes '-W' (valeur par défaut).
		ALPHABET.	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').
		ALPHANUM.	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.
			L'emplacement indiqué sur la ligne définissant la S.D. (zone SUITE à blanc) est reporté par défaut sur toutes ses lignes suite, au niveau de la documentation et de la génération.
			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.

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			ATTENTION : pour ORACLE, vous devez utiliser des valeurs numériques pour que la DECLARE SECTION soit générée correctement (avec les zones données et indicateurs).
28	10		ZONE COMPLEMENTAIRE
			Cette zone est de la forme DDSSRRRRRR avec :
			. DD = code Structure de Données,
			. SS = code Segment
			. RRRRRR = code Rubrique.
			Elle a trois significations :
			. NOMINAL, SYMBOLIC ou ACTUAL KEY suivant la variante, sur les matériels sur lesquels elle est nécessaire, sauf pour les fichiers en organisation 'V'.
			Vous devez définir la zone de WORKING correspondante : 1-DDSS-RRRRRR.
			Le positionnement de cette clé, ainsi que la lecture du fichier sont à programmer en langage structuré.
			. 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		Code de la clé dans l'enregistrement
			Clé dans l'enregistrement, obligatoire pour les fichiers en organisation indexée
30	6		NOM DU CODE ENREGISTREMENT
			C'est le code mnémorique de la Rubrique permettant d'identifier par ses valeurs les différents types d'enregistrements du fichier.
			Il se trouve obligatoirement dans la partie commune (code enregistrement 00).
			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.

Modification du début 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 BULL Gcos7 : 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 REPLISSAGE
1	2		INDICATIF 1
			Cette zone est saisissable, ce qui permet de copier des lignes d'un Programme vers un Ecran et vice versa.
		'O'	Ecran.
		'P'	Programme.
2	6		CODE PROGRAMME OU ECRAN
3	1		CODE ACTION
		'C'	Création de la ligne
		'M'	Modification de la ligne
		'A' ou 'D'	Annulation de la ligne
		'T'	Transfert de la ligne
		'B'	Annulation multiple
		'G'	Transfert d'un groupe de lignes
		'?'	Demande de documentation

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
		'X'	Mise à jour implicite sans transformation majuscule/ minuscule
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:
		01	Commentaire API COBOL
		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.
		01 90	RECEIVE-CONTROL (TANDEM)
		60	Est réservé à la ligne 'DATA DIVISION'.
		99 99	Est réservé à la ligne 'FILE SECTION'.
			Les clauses FD générées ne peuvent être ni modifiées ni annulées.
			Dans une Macro-Structure, le paragraphe généré peut être paramétré.
		9* FF	Réécriture de la clause FD pour le fichier FF.
6	3		NUMERO DE LIGNE
			ZONE NUMERIQUE PARAMETRABLE

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Il est recommandé de numéroter les lignes de 10 en 10 pour permettre d'en insérer de nouvelles par la suite.
	\$n	Dans une Macro-Structure, il est possible de paramétrer les deux premiers caractères du numéro de ligne.
7	66	INSTRUCTION A INSERER
		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).
	'S'	(ou tout mot commençant par 'S') : Entraîne la suppression de la section ou du paragraphe générés.
		Dans une Macro-Structure, cette zone peut être paramétrée.

Zones de travail et de lien (-W)

L'écran zones de travail et de lien (-W) 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 langage 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 VisualAge 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 DECLARATION dans la Description de l'Ecran).

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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.
		Pour un Programme Batch :
	'AA à ZZ'	Si le DEBUT DU NUMERO DE LIGNE est inférieur à '00'
	'0A à 0Z'	(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 :
		- 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 EMBLACEMENT DE LA DESCRIPTION supérieur ou égal au contenu de cette zone.
	'00 à 09'	Si le DEBUT DU NUMERO DE LIGNE est supérieur à '00',
	'1A à 19'	les données décrites dans l'écran seront générées en
	...	WORKING-STORAGE SECTION, après toutes les Structures
	'9A à 99'	de Données dont l'EMPLACEMENT DE LA DESCRIPTION (écran -CD) est inférieur au contenu de cette zone.
		Pour un Programme T.P. :
	'AA à 0Z'	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 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 peuvent être réservées pour des usages particuliers.
	'99'	Avec un NUMERO DE LIGNE = '999'

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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 paramétrer le contenu de cette zone.
4	1	CODE ACTION
	'C'	Création de la ligne
	'M'	Modification de la ligne
	'A' ou 'D'	Annulation de la ligne
	'T'	Transfert de la ligne
	'B'	Annulation multiple
	'G'	Transfert d'un groupe de lignes
	'?'	Demande de documentation
	'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
	'X'	Mise à jour implicite sans transformation majuscule/ minuscule
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, vous pouvez paramétrer les deux premiers caractères de cette zone de fin du numéro de ligne.
6	1	Type de déclaration
	blanc	Les données saisies dans les zones NIVEAU DE DECLARATION et DECLARATION DE DONNEES DE TRAVAIL sont générées telles qu'elles sont saisies
	'-'	Continuation d'un littéral
	'*'	Commentaire inséré dans le Programme généré. La dimension de la table n'est pas considérée comme un commentaire, voir la dernière zone de la ligne.
	'\$'	Cette valeur figurera dans le généré Cobol en colonne 7 et le reste de la ligne working sera repris tel quel

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'A'	Déclaration d'une Application eBusiness. Pour une documentation complète, consultez le "Guide d'utilisation de l'API COBOL".
	'F'	Appel d'une Structure de Données
		Si la ligne est créée à blanc, elle sera affichée avec un guide de saisie, les informations nécessaires correspondant exactement aux informations équivalentes de la ligne d'appel de Structures de Données pour les organisations 'W' et 'L' (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 sachant que dans un Ecran 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).
		Pour le 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
	'T'	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 d'édition
		Pour ces trois types d'appels, un contrôle sur l'existence dans le Dictionnaire est effectué
		Un chaînage particulier est réalisé interdisant l'annulation de cette Rubrique si les lignes où elle est appelée n'ont pas été préalablement annulées
		Si la Rubrique n'existe pas dans le Dictionnaire, un "warning" est envoyé par le système
		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
		Pour ces trois derniers types de déclaration, le nom-donnée déclaré doit être du type W-FFEE-RRRRRR avec :
		W = un préfixe quelconque (par exemple I, E ou S),

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			FFEE = un code Segment quelconque,
			RRRRRR = un code Rubrique existant dans le Dictionnaire
			Ce nom-donnée doit être cadré à gauche de la zone DECLARATION
			Module Dialogue (IMS)
		'M'	Déclaration d'un Ecran comme Moniteur
		'C'	Affectation d'un Ecran à un autre Ecran déclaré comme Moniteur (type 'M')
7	17		NIVEAU DE LA DECLARATION
			Nombre niveau (01, 05, ...) ou nom de section (LINKAGE SECTION).
			Dans une Macro-Structure, le niveau de la déclaration est paramétrable.
8	48		DECLARATION DES DONNEES DE TRAVAIL
			Dans tous les cas, on doit choisir les noms de zones (nom-données) en utilisant la normalisation VisualAge Pacbase.
			La structure de ces noms est W-FFEE-RRRRRR, avec:
			.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.
			Si cette normalisation est respectée, les références croisées Rubrique/Programme sont établies automatiquement.
			Dans le Programme généré, cette zone suit immédiatement la précédente.
			Dans une Macro-Structure, la déclaration de la donnée peut être paramétrée.
			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 contraire, le généré obtenu sera de la forme :
			03 FFEE-CORUB PICTURE X.
9	5		DIMENSION TABLE

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			ZONE NUMERIQUE PURE
			Déclenche la génération de trois indices :
			.IFFEEM initialisé à la valeur mentionnée,
			.IFFEEL initialisé à zéro, utilisable pour le remplissage de la table, donne le nombre de postes effectivement chargés.
			.IFFEER initialisé à zéro, utilisable pour la recherche 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'.

Chapitre 5. Langage Structuré

Description des traitements (-P)

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.

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. VA Pac 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 MULTI-BRANCHES

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. Vous avez 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 : Pour 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 opérands 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: PaaaaaaPG

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'

Vous pouvez vous débrancher sur l'écran 'CH : -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 0A
                                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 'OC7 PROVOQUE'                                99BL
* 90 90 M SPACE 7-DP00-DUMP
* 90 100 A 1 7-DP00-DUMP9

*** FIN ***
O: C1 CH: -P

```

DOCUMENTATION
 TRAITEMENTS PROGRAMME 1 P 2 TES001 test 001

P0*JFL.LURE.TES.972
 FONCTION: 3 OA

```

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
 0: C1 CH: -PG

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	2		INDICATIF 1
			Cette zone est saisissable, ce qui permet de copier des lignes d'un Programme vers un Ecran et vice versa.
		'O'	Ecran.
		'P'	Programme.
2	6		CODE PROGRAMME OU ECRAN
3	2		CODE FONCTION (OBLIGATOIRE)
			Le code doit être choisi en fonction de l'emplacement désiré, sachant que les fonctions automatiques occupent des emplacements déterminés.
			Dans une Macro-Structure, le code fonction peut être paramétré.
4	1		CADRAGE DES OPERANDES
			Permet d'afficher à l'écran la partie droite des opérandes.
		'BLANC'	Cadrage à gauche de la zone OPERANDES.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		'AUTRE'	Cadrage à droite de la zone OPERANDES.
			Cette option interdit toute mise à jour.
5	1		CADRAGE DE LA CONDITION
			Permet d'afficher à l'écran la partie droite de la condition.
		'BLANC'	Cadrage à gauche de la zone CONDITION.
		'AUTRE'	Cadrage à droite de la zone CONDITION. Cette option interdit toute mise à jour.
6	1		CODE ACTION
		'C'	Création de la ligne
		'M'	Modification de la ligne
		'A' ou 'D'	Annulation de la ligne
		'T'	Transfert de la ligne
		'B'	Annulation multiple
		'G'	Transfert d'un groupe de lignes
		'?'	Demande de documentation
		'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
		'X'	Mise à jour implicite sans transformation majuscule/ minuscule
7	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.
8	3		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.
		\$n	Dans une Macro-Structure, il est possible de paramétrer les deux premiers caractères du numéro de ligne.
			TRAITEMENT
			L'écriture des traitements spécifiques est basée sur l'utilisation d'un opérateur, suivi des opérandes correspondants.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		L'écriture des opérands peut se poursuivre sur plusieurs lignes. Dans ce cas, l'opérateur ne doit pas être précisé sur les lignes suite.
		De manière générale, la ponctuation est gérée automatiquement, essentiellement en fonction de la hiérarchisation des traitements. Vous pouvez cependant prendre en charge cette ponctuation.
		Dans une Macro-Structure, l'opérateur peut être paramétré.
9	3	OPERATEURS DE TRAITEMENTS COURANTS
	'N '	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.
	'* '	Commentaire.
		Cet opérateur doit figurer sur chaque ligne de commentaire (même si elles se suivent).
		Voir aussi la zone OPTION INSERTION COMMENTAIRES dans la Définition de la Bibliothèque.
	'M '	Transfert du premier opérande dans le(s) suivant(s).
	'MA '	Transfert du premier opérande dans le(s) suivant(s) en utilisant l'ordre COBOL 'MOVE ALL'.
	'MC '	Transfert du premier opérande dans le suivant en utilisant l'ordre COBOL 'MOVE CORRESPONDING'
	'MCI'	Transfert entre Rubriques dépendantes identiques en utilisant l'ordre COBOL 'MOVE'.
		Le premier opérande est la zone groupe source du transfert et le deuxième la zone groupe cible. Le transfert est effectué pour toute Rubrique élémentaire de même code dans les deux zones groupe. le critère de rapprochement correspond à la chaîne de caractères située derrière le premier tiret ('-').
		Pour activer cet opérateur, lancez la procédure GPMC (voir le manuel "Procédures du Développeur").
	'MF '	Génération de l'ordre COBOL 'MOVE FUNCTION'. La fonction, indiquée en première opérande, est une des fonctions intrinsèques du COBOL (ex : UPPER-CASE). Référez-vous à la documentation COBOL pour la liste de ces fonctions et leur syntaxe.

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'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.
		'C '	Calcul.
			Dans les opérandes la zone RESULTAT 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.
		'A '	Addition du premier opérande dans le(s) suivant(s).
		'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 OPERANDES. (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 OPERANDES.
			Si la (sous-)fonction est de type 'TT' 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.
			Dialogue et Pacbench C/S :
			Débranchement à la fin du traitement des catégories.
			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.
			Dialogue et Pacbench C/S :
			Passage à l'occurrence suivante de la catégorie en cours ou à la catégorie suivante.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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.
	'EXA'	Génère l'ordre COBOL 'EXAMINE' de la zone placée en opérande, éventuellement suivie de clauses complémentaires (TALLYING...); voir la syntaxe COBOL.
		Note : pour une génération COBOL II, utiliser à la place l'opérateur INS.
	'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.
		batch : Coder la valeur SUP sur la première ligne de la sous-fonction pour la supprimer.
	'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 VA Pac.
		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.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'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)
	'EVF'	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 OPERANDES
	'UNS'	Génère l'ordre COBOL 'UNSTRING' devant les paramètres indiqués dans la zone OPERANDES
		OPERATEURS PACBENCH C/S ET DIALOGUE
		DEBRANCHEMENT
	'GF'	Débranchement à la fin de la sous-fonction automatique où la ligne est insérée.
		NOTE POUR ECRAN C/S :
		Utilisable uniquement en fonctions 20, 25, 35 et 60.
		NOTE POUR COMPOSANT APPLICATIF :
		Avec ENDV dans la zone OPERANDES, fin de traitement de Vue Logique.
	'GFR'	Abandon des traitements de réception (débranchement à l'étiquette END-OF-RECEPTION).

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'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 DE TRAITEMENTS (Spécifique Composant Applicatif)
	'XT'	Appel d'un traitement élémentaire sur Vue Logique ou sur Segment.
		Pour plus de détails, consultez le manuel "Pacbench C/S - Services Applicatifs et Client TUI", chapitre "Composant Applicatif", sous-Chapitre "Description des Traitements (-P)", Section "Opérateurs".
		APPEL (Spécifique Ecran C/S et Ecran)
	'OTP'	Appel immédiat d'un nom externe.
	'OSC'	Appel de l'écran indiqué en opérande.
	'OSD'	Appel de l'écran indiqué en opérande après exécution des traitements de réception.
		LECTURE ET ECRITURE
		Le code du Segment est indiqué en OPERANDES.
	'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.
		NOTE POUR DIALOGUE :

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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), VA Pac 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), VA Pac génère automatiquement une étiquette F80-MV00-aa.
	'Y'	Opérateur spécifique au Composant Applicatif Génération de l'étiquette de la fonction automatique lorsque celle-ci a été remplacée par traitement spécifique (*R).
	'ERU'	Positionnement d'Erreur Utilisateur.
		La zone OPERANDES doit contenir :
		. code de l'erreur géré par l'utilisateur (4 car.),
		. un blanc,
		. éventuellement le code de la Rubrique sur six positions.
		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.
		Cet opérateur ne peut pas être utilisé sur une Rubrique répétée.
	'ERR'	Positionnement d'erreur standard sur Rubrique.
		La zone OPERANDES doit contenir :
		. 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. Pour une Rubrique répétée, on indique l'indice en clair sur lequel porte l'erreur. Exemple : CORUB (2)

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'ERL'	Spécifique au Composant Applicatif associé à une application graphique (mode Dossier ou mono-vue). Cet opérateur permet de positionner une erreur sur une demande de LOCK ou UNLOCK par un client graphique, pour une occurrence déjà verrouillée ou non verrouillée, respectivement.
		Pour plus de détails, consultez le manuel "Pacbench C/S - Services Applicatifs".
		L'utilisation des libellés d'erreur avec le module Langage Structuré est détaillée dans le manuel de référence correspondant.
		OPERATEURS DE TRAITEMENT BATCH
		Ces opérateurs ne sont pris en compte que dans la génération de Programmes batch.
		Ils assurent les ordres d'entrée-sortie des fichiers en organisation séquentielle, séquentielle indexée ou dynamique.
		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.
		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.). Vous aurez la charge de déterminer l'action à entreprendre en fonction de la valeur de cette variable.
		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'.
	'OPE'	Ouverture.
	'CLO'	Fermeture.
	'R '	Lecture.
	'W '	Ecriture.
	'RW '	Réécriture.
	'TRI'	Tri.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'RN '	Read Next.
	'STA'	Start.
	'DEL'	Delete.
		Pour les opérateurs 'OPE', 'CLO', 'R ', 'RN ' et 'DEL' le code S.D.-programme (2c) est le seul opérande.
		Pour l'écriture et la réécriture, la zone OPERANDES doit contenir le code enregistrement (FFEE) suivi éventuellement d'options telles que FROM ou AFTER.
		Pour le START, on indique le code S.D-Programme, suivi éventuellement du positionnement de la clé au départ :
		Exemple : STA FF NOT < FF00-START génère
		MOVE 0 TO IK
		START FF-FICHER KEY IS NOT < FF00-START
		INVALID KEY MOVE 1 TO IK.
		Pour le tri les opérandes sont les paramètres suivant le verbe SORT.
	'E'	Repère un libellé d'erreur utilisateur (Langage batch)
		La zone OPERANDES doit contenir :
		. 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.
		La deuxième partie du libellé doit être codifiée dans la zone CONDITION (32 positions).
		OPERATEURS DE TRAITEMENT DE DATE
		TRAITEMENT DE DATE
	'ADT'	Appel de la date machine.
		La date sera fournie, quel que soit le matériel, sous la forme an, mois, jour dans la constante VA Pac '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.
		Remarque : en COBOL II et COBOL 85, 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 initialisé à '20'.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'ADC'	Fournit la date machine sous la forme CCAAMMJJ où 'CC' représente le siècle.
			Remarque : en COBOL II et COBOL 85, 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 initialisé à '20'.
			Formatage de la date :
		'AD'	Si la condition des opérateurs de traitement de date nécessite plusieurs lignes, les opérands doivent être indiqués avant la suite de la condition (sans opérateur ou sur des lignes commentaire).
			Une date peut être formatée de différentes façons :
			Les opérands utilisables sont 'XY CORUB1 CORUB2' où 'X' et 'Y' représentent les codes du tableau ci-dessous.
			Code : I
			Format généré : AN, MOIS, JOUR
			Code : D
			Format généré : JOUR, MOIS, AN ou MOIS, JOUR selon le type de format demandé dans le Dictionnaire.
			Code : E
			Format généré : JOUR/MOIS/AN ou MOIS/JOUR/AN (Etendu)
			Code : S
			Format généré : SIECLE, AN, MOIS, JOUR
			Code : C
			Format généré : MOIS, JOUR, SIECLE, AN ou JOUR, MOIS, SIECLE, AN
			Code : M
			Format généré : MOIS/JOUR/SIECLE AN ou JOUR/MOIS/SIECLE AN
			Code : G Format généré : SIECLE AN - MOIS - JOUR
			EXEMPLE :

NUM	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		Pour transformer une date de format 'I' en date de format 'D', il convient d'indiquer 'AD' dans la zone OPERATEUR, 'ID CORUB1 CORUB2' dans la zone OPERANDES, CORUB1 étant la Rubrique contenant une date en an, mois, jour (il est possible d'utiliser la constante VA Pac 'DATOR' si besoin), et CORUB2 étant la Rubrique date transformée en jour, mois, an.
		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 -GO du Programme, le paramètre DATPRO = ffss, où ffss est le code fonction-sous fonction choisi.
	'AD0'	Alimentation du siècle à partir d'une zone DAT-CTY initialisée à '20' 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.
		CALCUL SUR LES DATES
	'DAD'	Calcul du nombre de jours entre deux dates. Dans la zone OPERANDE, vous devez saisir :
		xy ffnn-date1 ssnn-date2
		"x" correspondant au format de "date1" (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		"y" correspondant au format de "date2" (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		Le résultat de la comparaison est mis dans la zone NUM-DAYS, qui est automatiquement déclarée en Working.
	'DAO'	Addition ou soustraction de jours à une date. Dans la zone OPERANDE, vous devez saisir :
		x snumber ffnn-date1
		"x" correspondant au format de la date (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si les Rubriques ont un format date;
		"s" correspondant au signe "+" pour une addition ou "-" pour une soustraction

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		"number" correspondant au nombre à ajouter/retrancher à la date. Il peut être un entier ou une zone de travail.
		"ffrn-date1" correspondant à la zone date en entrée et en sortie.
		Pour alimenter le résultat dans d'autres dates, en plus de date1, vous devez saisir ces autres dates (de format identique à date1) sur une ligne suite sans opérateur.
		TRAITEMENT DE DATE (Dialogue)
	'AD6'	ADT + ADI 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.
	'AD8'	ADT + ADI + ADE 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.
		TRAITEMENT DE DATE (Batch)
	ADI	Inversion d'une date (tous modules sauf Dialogue): les deux premiers et les deux derniers caractères sont inversés. Les deux opérandes sont supposés avoir une longueur de six caractères. Le deuxième opérande n'est pas obligatoire, dans ce cas, le premier opérande récupère la date inversée.
	ADS	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.
	ADE	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. doit avoir une longueur de huit caractères.
	ADM	Mise en forme d'une date avec siècle par insertion de slash ('/') ; (tous modules sauf Dialogue).
		TRAITEMENT DE L'HEURE
	'TIM'	Fournit l'heure sous la forme HHMMSS à partir de la zone EIBTIME pour CICS, et TIME pour les autres matériels.
	'TIF'	Transforme le format de l'heure HHMMSS en HH:MM:SS.
		OPERATEURS DE COMMUNICATION
		NOTE: Non opérationnels avec le module Dialogue.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Ils assurent les liaisons entre un Programme COBOL et les 'UNITES DE COMMUNICATION'.
		'ENA'	ENABLE connexion de l'unité.
		'DSB'	DISABLE déconnexion de l'unité.
		'RE '	RECEIVE réception d'un message.
		'SD '	SEND émission d'un message.
			OPERATEURS DES S.G.B.D.
			OPERATEURS CODASYL
			Ils se caractérisent par une première lettre 'B' suivie de deux caractères, définis ci-dessous, spécifiques de l'instruction.
		'RY'	READY
		'FH'	FINISH
		'FD'	FIND
		'G '	GET
		'ER'	ERASE
		'DT'	DISCONNECT --- FROM ---
		'CT'	CONNECT --- TO ---
		'MD'	MODIFY
		'ST'	STORE
			OPERATEURS POUR MONITEURS T.P.
			Ces opérateurs ne sont pas utilisables pour les descriptions de traitements d'Ecran avec le module Dialogue.
			OPERATEURS 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.
			OPERATEURS CICS COMMAND LEVEL
		'EXEC'	EXEC CICS opérandes END-EXEC.
			OPERATEURS API COBOL et PAF/DAF

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'EXS'	Appel d'un Gestionnaire de Dossier. Cet opérateur est utilisé pour écrire une demande de service sur un Dossier.
		Pour plus de détails, voir le "Guide d'utilisation de l'API COBOL".
	'EXP'	Cet opérateur est utilisé pour activer une requête d'accès à la base VA Pac à l'aide du module PAF (Pacbase Access Facility).
		Pour plus de détails, voir le manuel "Pacbase Access Facility" ou le manuel "DSMS Access Facility".
		OPERATEURS SQL
		Les opérateurs SQL sont documentés dans le manuel "Description des Bases de Données SQL", chapitre "Accès SQL".
	SCC	Ordre CONNECT ou son équivalent.
	SDC	Ordre DISCONNECT ou son équivalent.
	SCO	Ordre COMMIT.
	SRO	Ordre ROLLBACK.
	SWH	Ordre WHENEVER.
		La syntaxe doit être la suivante :
		SCC ccccc d
		SDC ccccc d r
		SCO ccccc d
		SRO ccccc d
		SWH instruction
		cccccc : code VA Pac du Bloc (sur 6 caractères).
		d : valeur 2 si base distribuée (ex : Oracle Sybase).
		r : valeur R pour choisir le DISCONNECT avec ROLLBACK.
		Règles :
		- 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.
		Génération :

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			- 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.
			La génération du END-EXEC est automatique et, pour le générateur batch, le END-EXEC est toujours suivi d'un point.
		SQL	Pour batch uniquement, l'ordre SQL obtenu est standard avec les possibilités de personnalisation introduites dans l'écran -GG du Segment.
			La personnalisation des accès SQL est documentée dans le manuel "Description des Bases de Données SQL", chapitre "Accès SQL", sous-chapitre "Personnalisation des accès SQL".
			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.
			L'opérateur SQL doit avoir la syntaxe suivante :
			OPE I OPERANDES
			SQL I FFSS FBSS OS OP (format 1)
			SQL I FFSS OS OP (format 2)
			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 S FBSS GG.
			OP : ordre particulier décrit dans S FBSS GG.
			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.
			Les organisations SQL retenues sont :
			C (Interel)
			M (Datacom)

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			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)
			Pour certaines organisations, il faut avoir la référence au type de Bloc (H et 9 par exemple).
			Restrictions :
			- 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.
			Limite :
			Dans les recouvrements programme-macro et macro-macro, les lignes générées obtenues par le S FFSS GG ne peuvent pas être supprimées.
			OPERATEUR RELATIONNEL
		EXQ	EXEC SQL opérandes END-EXEC.
10	32		OPERANDES
			Le paramétrage de cette zone dépend de l'opérateur qui lui est associé.
			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.
			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.
			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, VA Pac assurera la continuité au niveau de la génération.

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			STRUCTURE
			Toute structure (fonction ou sous-fonction) a un nombre niveau et un type, auxquels on associe éventuellement une condition, une variable ou une valeur.
			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.
			Les valeurs par défaut concernant le niveau et le type sont prises sur la ligne titre de la (sous-) fonction.
			Les fonctions constituent les structures de plus haut niveau (05).
11	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 du Programme jusqu'à ce que l'on rencontre une structure de niveau égal ou supérieur.
		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.
12	2		TYPE DE STRUCTURE OU DE CONDITION
			Sur la première ligne de Définition d'un bloc de traitement, indique le type du bloc conformément aux règles de la programmation structurée.
			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.
			Sur les autres lignes définit une suite de condition.
		'BL'	Structure de type 'BLOCK'.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Valeur par défaut pour toute Structure non conditionnée.
	'IT'	Structure de type 'IF THEN'.
		Exécutée si son conditionnement est satisfait. Valeur par défaut pour toute structure conditionnée.
	'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.
	'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, 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.
	'DW'	Structure de type 'DO WHILE'.
		Interdite au niveau fonction et au niveau 99.
		Structure exécutée de façon répétitive tant que son conditionnement est satisfait.
	'DU'	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.
	'DO'	Structure de type 'DO' (structure de type boucle).
		Interdite au niveau fonction et 99.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			Structure exécutée de façon répétitive entre 2 bornes (précisées dans la zone CONDITION) en fonction d'un pas positif ou négatif. Les bornes sont obligatoires, alors que le pas est facultatif.
			Les bornes peuvent être soit une valeur, soit un nombre (signé ou non) et doivent être séparées par un espace.
			Les paramètres doivent être renseignés, sur la première ligne, dans l'ordre suivant :
			Borne inférieure (positive), borne supérieure et le pas d'incrémentation de l'indice.
			VisualAge Pacbase génère automatiquement un indice JffnnR, 'ffnn' représentant le code (sous-)fonction.
			Pour générer un autre indice, vous devez saisir en première position de la zone CONDITION :
			I=indice borne1 borne2 pas
			où 'indice' correspond à une zone de travail numérique.
		'OR'	Suite de conditionnement associée aux lignes précédentes par un 'OU' logique.
		'AN'	Suite de conditionnement associée aux lignes précédentes par un 'ET' logique.
			NB : les parenthèses regroupant des termes dans une condition doivent être indiquées dans le texte de la condition.
			Dans une Macro-Structure le type de structure ou de condition ne peut pas être paramétré.
		'WH'	Vous pouvez utiliser des ordres COBOL derrière un ordre EVA ou SEA.
			Chacun des ordres indique le traitement à effectuer pour une condition particulière.
			Dans l'exemple suivant, le traitement t1 est effectué lorsque la condition c1 est remplie.
			EVA -----
			---t1--- 99WH ---c1---
			---t2--- 99WH ---c2---
			---t3--- 99WH ---c3---
			CONDITIONS RELATIVES A DES DATES
		'DC'	Exécution des traitements en fonction de la comparaison entre deux dates.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Vous devez saisir dans la zone CONDITION :
		xy ffnn-date1 oper ssnn-date2
		où
		'x' correspond au format de 'date1' (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		'y' correspond au format de 'date2' (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		'oper' correspond au type de comparaison. Il peut prendre les valeurs suivantes : >, <, >=, <=, NOT>, NOT<, NOT=.
		Si cette condition est indiquée sur une sous-fonction, le test de comparaison sera suivi du traitement suivant :
		NEXT SENTENCE ELSE GO TO Fffss-FN.
	'DV'	Exécution des traitements si la date est valide.
		Vous devez saisir dans la zone CONDITION :
		x ffnn-date1
		où
		'x' correspond au format de la date (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		'ffnn-date1' correspond à la Rubrique date à contrôler
		Si cette condition est indiquée sur un niveau 99, la ligne suivante sera générée pour que les traitements dépendants soient exécutés : IF EN-PRE='1'
		Si cette condition est indiquée au niveau d'une sous-fonction 'ffss', les lignes suivantes seront générées pour que les traitements dépendants soient exécutés :
		IF EN-PRE='1'
		NEXT SENTENCE ELSE GO TO Fffss-FN.
	'DI'	Exécution des traitements si la date est invalide.
		Vous devez saisir dans la zone CONDITION :
		x ffnn-date1
		où
		'x' correspond au format de la date (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		'ffnn-date1' correspond à la Rubrique date à contrôler
		Si cette condition est indiquée sur un niveau 99, la ligne suivante sera générée pour que les traitements dépendants soient exécutés :
		IF EN-PRE>'1'
		Si cette condition est indiquée au niveau d'une sous-fonction 'ffss', les lignes suivantes seront générées pour que les traitements dépendants soient exécutés :
		IF EN-PRE>'1'
		NEXT SENTENCE ELSE GO TO Fffss-FN.
		DIALOGUE & PACBENCH C/S :
	'*A'	Insertion de la sous-fonction avant une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.
	'*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.
	'*R'	Remplacement d'une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.
		Pour plus de détails concernant ces trois types, se reporter aux manuels "Dialogue" et "Pacbench C/S - Services Applicatifs".
		Les deux types suivants sont réservés au module Pacbench C/S (composants serveur uniquement) :
	'*C'	Insertion ou remplacement des traitements sur serveur ou vue logique. Il ne peut être indiqué que sur un niveau 05.
	'*B'	Insertion dans les traitements élémentaires appelés par PERFORM.
		Pour plus de détails concernant ces deux types, se reporter au manuel "Services Applicatifs et clients TUI", chapitre "Composant Applicatif", sous-chapitre "Description des traitements générés".
13	28	CONDITION D'EXECUTION
		Exprimé sans 'IF', ni 'AND', ni 'OR', ni 'GO TO', ni point.

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			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 variable sujet de l'alternative doit être indiqué sur une seule ligne.
			Pour les Structures dépendant d'un 'CASE OF', la condition d'exécution représente une valeur possible de la variable sujet du 'CASE OF'.
			Dans une Macro-Structure, la condition d'exécution peut être paramétrée.

Chapitre 6. Titres, conditions et niveaux des traitements

Titres et conditions des traitements (-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 voir 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,
CH : P.....CD,
- L'appel de Segments dans les Programmes conversation- nels ou Ecrans,
CH : 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 VisualAge Pacbase en ajoutant certains éléments spécifiques du Programme. Elle est mise en oeuvre par :

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

CH : P.....CP ou O.....CP,

- L'appel direct de Langage Structuré,

CH : 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, vous pouvez 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 :

'PaaaaaTCffxx<nn' ou

'Paaaaa<nnTCffxx' pour les Programmes,

'OsssssTCffxx<nn' ou

'Oaaaaa<nnTCffxx' pour les Ecrans.

Avec :

.'aaaaaa' = code Programme
. 'ssssss' = code Ecran
. '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', entrez le choix :

Ppgm001TC

Si, pour éviter d'avoir une image trop détaillée du Programme, vous souhaitez l'affichage des titres et conditionnements jusqu'au niveau 15, à partir de la fonction 'ff', sous-fonction 'xx', vous devez entrer le choix suivant :

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

REPERAGE DES LIGNES AFFICHEES

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

1 - GENERATION AUTOMATIQUE VA Pac :

Dans ce cas, les lignes affichées proviennent des lignes '-CD' pour les Programmes ou '-CS' pour les Ecrans. Elles sont repérées par un '.' dans le code action.

2 - APPELS DE MACRO-STRUCTURES :

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

3 - LANGAGE STRUCTURE 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, vous devez effacer ce code action.

TRAITEMENTS SPECIFIQUES :

Vous pouvez créer, modifier ou annuler des titres de (sous-)fonctions introduits grâce au langage structuré.

Pour cela, vous disposez des codes actions 'C', 'M', 'A' ou 'blanc'.

Vous pouvez modifier le titre, le niveau, le type et le conditionnement d'une (sous-)fonction.

DEBRANCHEMENT SUR UNE (SOUS-)FONCTION AFFICHEE

Vous avez la possibilité de vous débrancher sur l'écran '-P' correspondant à une ligne affichée. Pour cela, déplacez le curseur sur la ligne concernée et utilisez la touche fonction réservée à cet effet (en standard PF10).

De même, vous pouvez vous 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, vous pouvez 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 REQUISES

Le Programme ou l'Ecran 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.

TRAITEMENTS PROGRAMME 1 F020MA MISE A JOUR FOURNISSEURS

```

A FOSS NLG 0 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
0: C1 CH: -TC
    
```

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	6		CODE PROGRAMME OU ECRAN
2	1		CODE ACTION
		'C'	Création de la ligne
		'M'	Modification de la ligne
		'A' ou 'D'	Annulation de la ligne
		'T'	Transfert de la ligne
		'B'	Annulation multiple
		'G'	Transfert d'un groupe de lignes
		'?'	Demande de documentation
		'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
		'X'	Mise à jour implicite sans transformation majuscule/ minuscule
3	2		CODE FONCTION

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Le code doit être choisi en fonction de l'emplacement désiré, sachant que les fonctions automatiques occupent des emplacements déterminés.
		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	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.
	\$n	Dans une Macro-Structure, il est possible de paramétrer les deux premiers caractères du numéro de ligne.
6	1	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 du Programme jusqu'à ce que l'on rencontre une structure de niveau égal ou supérieur.
	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.

NUM	MLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		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.
9	2		TYPE DE STRUCTURE OU DE CONDITION
			Sur la première ligne de Définition d'un bloc de traitement, indique le type du bloc conformément aux règles de la programmation structurée.
			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.
			Sur les autres lignes définit une suite de condition.
		'BL'	Structure de type 'BLOCK'.
			Valeur par défaut pour toute Structure non conditionnée.
		'IT'	Structure de type 'IF THEN'.
			Exécutée si son conditionnement est satisfait. Valeur par défaut pour toute structure conditionnée.
		'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.
		'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, 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.
		'DW'	Structure de type 'DO WHILE'.
			Interdite au niveau fonction et au niveau 99.

NUM	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LONG	VALEUR	REPLISSAGE
		Structure exécutée de façon répétitive tant que son conditionnement est satisfait.
	'DU'	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.
	'DO'	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 CONDITION) en fonction d'un pas positif ou négatif. Les bornes sont obligatoires, alors que le pas est facultatif.
		Les bornes peuvent être soit une valeur, soit un nombre (signé ou non) et doivent être séparées par un espace.
		Les paramètres doivent être renseignés, sur la première ligne, dans l'ordre suivant :
		Borne inférieure (positive), borne supérieure et le pas d'incrément de l'indice.
		VisualAge Pacbase génère automatiquement un indice JffnnR, 'ffnn' représentant le code (sous-)fonction.
		Pour générer un autre indice, vous devez saisir en première position de la zone CONDITION :
		I=indice borne1 borne2 pas
		où 'indice' correspond à une zone de travail numérique.
	'OR'	Suite de conditionnement associée aux lignes précédentes par un 'OU' logique.
	'AN'	Suite de conditionnement associée aux lignes précédentes par un 'ET' logique.
		NB : les parenthèses regroupant des termes dans une condition doivent être indiquées dans le texte de la condition.
		Dans une Macro-Structure le type de structure ou de condition ne peut pas être paramétré.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'WH'	Vous pouvez utiliser des ordres COBOL derrière un ordre EVA ou SEA.
		Chacun des ordres indique le traitement à effectuer pour une condition particulière.
		Dans l'exemple suivant, le traitement t1 est effectué lorsque la condition c1 est remplie.
		EVA -----
		---t1--- 99WH ---c1---
		---t2--- 99WH ---c2---
		---t3--- 99WH ---c3---
		CONDITIONS RELATIVES A DES DATES
	'DC'	Exécution des traitements en fonction de la comparaison entre deux dates.
		Vous devez saisir dans la zone CONDITION :
		xy ffnn-date1 oper ssnn-date2
		où
		'x' correspond au format de 'date1' (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		'y' correspond au format de 'date2' (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		'oper' correspond au type de comparaison. Il peut prendre les valeurs suivantes : >, <, >=, <=, NOT>, NOT<, NOT=.
		Si cette condition est indiquée sur une sous-fonction, le test de comparaison sera suivi du traitement suivant :
		NEXT SENTENCE ELSE GO TO Fffss-FN.
	'DV'	Exécution des traitements si la date est valide.
		Vous devez saisir dans la zone CONDITION :
		x ffnn-date1
		où
		'x' correspond au format de la date (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		'ffnn-date1' correspond à la Rubrique date à contrôler

NUM	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LN	VALEUR	REPLISSAGE
		Si cette condition est indiquée sur un niveau 99, la ligne suivante sera générée pour que les traitements dépendants soient exécutés : IF EN-PRE='1'
		Si cette condition est indiquée au niveau d'une sous-fonction 'ffss', les lignes suivantes seront générées pour que les traitements dépendants soient exécutés :
		IF EN-PRE='1'
		NEXT SENTENCE ELSE GO TO Fffss-FN.
	'DI'	Exécution des traitements si la date est invalide.
		Vous devez saisir dans la zone CONDITION :
		x ffnn-date1
		où
		'x' correspond au format de la date (facultatif). Si vous ne l'indiquez pas, le format sera extrait du référentiel, si la Rubrique a un format date.
		'ffnn-date1' correspond à la Rubrique date à contrôler
		Si cette condition est indiquée sur un niveau 99, la ligne suivante sera générée pour que les traitements dépendants soient exécutés :
		IF EN-PRE>'1'
		Si cette condition est indiquée au niveau d'une sous-fonction 'ffss', les lignes suivantes seront générées pour que les traitements dépendants soient exécutés :
		IF EN-PRE>'1'
		NEXT SENTENCE ELSE GO TO Fffss-FN.
		DIALOGUE & PACBENCH C/S :
	'*A'	Insertion de la sous-fonction avant une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.
	'*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.
	'*R'	Remplacement d'une sous-fonction automatique identifiée par la Rubrique ou le Segment qu'elle traite.
		Pour plus de détails concernant ces trois types, se reporter aux manuels "Dialogue" et "Pacbench C/S - Services Applicatifs".

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Les deux types suivants sont réservés au module Pacbench C/S (composants serveur uniquement) :
		'*C'	Insertion ou remplacement des traitements sur serveur ou vue logique. Il ne peut être indiqué que sur un niveau 05.
		'*B'	Insertion dans les traitements élémentaires appelés par PERFORM.
			Pour plus de détails concernant ces deux types, se reporter au manuel "Services Applicatifs et clients TUI", chapitre "Composant Applicatif", sous-chapitre "Description des traitements générés".
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 variable sujet de l'alternative doit être indiqué sur une seule ligne.
			Pour les Structures dépendant d'un 'CASE OF', la condition d'exécution représente une valeur possible de la variable sujet du 'CASE OF'.
			Dans une Macro-Structure, la condition d'exécution peut être paramétrée.

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

Vous avez la possibilité de vous débrancher sur l'écran 'CH : -P' correspondant à une ligne affichée. Pour cela, placez le curseur sur la ligne concernée et utilisez la touche fonction réservée à cet effet (en standard PF10).

CONDITIONS REQUISES

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.

VALIDATION MATERIEL IBM

P0*AG.LILI.IBM.847

TRAITEMENTS PROGRAMME F020MA MISE A JOUR FOURNISSEURS

. FOSF	. TITRE	. NIVEAU	. TY	. SOURCE.BIBLI
. 01	. INITIALISATIONS	. 05	. BL	.
. 05	. LECTURE FICH. SEQ. SS RUPTURE	. 05	. BL	.
. 05FBAAMA11.*VAL
. 10	. LECTURE FICH. SEQ. A RUPTURE	. 05	. BL	.
. 10FBAAMA11.*VAL
. 19	. VIDAGE DE LA TABLE	. 05	. IT	.*VAL
. 20	. FIN DE TRAITEMENT	. 05	. IT	.
. 22	. CALCUL DERNIERS ENREG.	. 05	. BL	.
. 24AAMA11.*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

0: C1 CH: -TO

Chapitre 7. Répertoire d'accès

Accès aux Ecrans

LISTE DES PROGRAMMES

- LCPaaaaaa
Liste des Programmes par code (à partir du Programme 'aaaaaa').
- LNPaaaaaaaa
Liste des Programmes par libellé (à partir du libellé 'aaaaaaaa') (choix sensible aux majuscules/minuscules).
- 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'.
- PaaaaaaGCbbb (MAJ)
Commentaires du Programme 'aaaaaa' (à partir de la ligne 'bbb').
- PaaaaaaGObbb (MAJ)
Options de génération 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').
- PaaaaaaCRrrrrrr (MAJ)
Instances reliées au Programme 'aaaaaa' par la Relation Utilisateur 'rrrrrr'.
- PaaaaaaCDbb (MAJ)
Appels de Structures de Données dans le Programme 'aaaaaa' (à partir de la Structure de Données 'bb').
- 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').

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

Tous les éléments expliqués entre parenthèses sont facultatifs.

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
AACA10	AACA10	CADRAGE D'UNE ZONE SAISIE	N				AACA10	PB	P	*CEN
AACA11	AACA11	APPEL CADRAGE D'UNE ZONE	N				AACA11	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

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

```

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.

Codes action en TP

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 la ligne 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 guide de "l'Interface Utilisateur Mode Caractère".

Edition et génération

- LCP

Liste des Programmes par code

C1 : Edition de la ligne de Définition de chaque Programme, sans les mots-clés.

C2 : Edition de la ligne de Définition de chaque Programme, avec les mots-clés.

- LEP

Liste des Programmes par nom externe.

- 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 guide de "l'Interface Utilisateur Mode Caractère").

- LNP

Liste des Programmes par nom.

- LTP

Liste des Programmes par type.

- DCP

Description du Programme dont le code est indiqué dans la zone ENTITE.

Description de tous les Programmes si la zone n'est pas renseignée :

C1 : Edition de toutes les lignes de description du Programme sans les Textes ventilés.

C2 : Edition de toutes les lignes de description du Programme avec les Textes ventilés.

- DSP

Description du Programme sélectionné issu de Reverse engineering.

- GCP

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

Lors de la phase de génération, les Segments des Structures de Données appelées dans le Programme pourront contenir jusqu'à 9999 Rubriques. Un message d'anomalie sera édité dans le compte-rendu de génération si cette limite est dépassée.

- GSP

Génération du programme sélectionné issu de Reverse engineering (comportant des lignes -SC).

- FLP

Contrôle du flot des Programmes. Vous pouvez préciser l'environnement, des options cartes de contrôle et éventuellement des paramètres.

- FSP

Contrôle du flot des Programmes issus de Reverse Engineering.

Chapitre 8. Description d'un Programme généré

Introduction

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 !  
-----
```

Identification Division

Vous avez la possibilité de modifier cette division dans l'écran 'CH : -B'.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PJJPLS. PJJPLS  
AUTHOR. COMMANDES. PJJPLS  
DATE-COMPILED. 12/04/06. PJJPLS
```

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 Bull, 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 où XX, NN, CORUB ont été définis dans les appels de S.D.

En effet, cette clé figure généralement dans un fichier en entrée, sinon il vous appartient de la définir et de la gérer.

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

```
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-FICHIER    ASSIGN    UT-S-SOR01.    PJJPLS
    SELECT    LA-FICHIER    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-FICHIER    ASSIGN    DA-I-LB        PJJPLS
RECORD KEY IS                                     LB00-MOIS.   PJJPLS
    SELECT    LK-FICHIER    ASSIGN    UT-S-TRI01.    PJJPLS
    SELECT    TR-FICHIER    ASSIGN TO ENT01          PJJPLS
ORGANIZATION INDEXED                                PJJPLS
FILE STATUS IS                                     1-TR00-STATUS PJJPLS
RECORD KEY IS                                     TR00-CLECD.  PJJPLS
```

Data Division File Section

Vous ne pouvez 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.

DATA DIVISION.				PJJPLS
FILE SECTION.				PJJPLS
FD		KL-FICHIER		PJJPLS
	BLOCK		00000 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-FICHIER		PJJPLS
	BLOCK		00000 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-FICHIER	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
	10	LB20-REDCAN PICTURE X(6).	PJJPLS

SD	10	LB20-FILLER PICTURE X(19).	PJJPLS
		LK-FICHIER	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-FICHIER	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
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

Début de 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 :

NUGNA : Numéro de session de génération du programme,
APPLI : Code de la bibliothèque,
DATGN : Date de la génération (JJ/MM/AA si langue de
l'utilisateur = 'F', ou MM/JJ/AA sinon),
PROGR : Code du programme dans la bibliothèque,
CODUTI : Code de l'utilisateur ayant demandé la génération,
TIMGN : Heure de la génération,
PROGE : Nom externe du programme,
COBASE : Code de la base,
DATGNC : Date de la génération avec siècle (JJ/MM/SSAA si
langue de l'utilisateur = 'F', ou MM/JJ/SSAA sinon)
RELEAS : Version du générateur,
DATGE : Date du générateur,
DATSQ : Date du squelette.

La variable DATCE est composée d'une zone CENTUR contenant la valeur du siècle courant, initialisée à '20', et d'une zone date (DATOR) non initialisée dans laquelle vous pourrez 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 la valeur saisie dans la zone ANNEE DU FENETRAGE de la fiche Bibliothèque, la zone CENTUR est automatiquement initialisée à '20'.

```
WORKING-STORAGE SECTION.                                PJJPLS
*SD: WR BIB: WR SEL: 02_____ FORM: I DESC: 2 NIV: 2 ORG: _ SS: _ 7BB010
01                WR00.                                  PJJPLS
                   WR02.                                  PJJPLS
                   WR02-DAT1.                             PJJPLS
                   WR02-DAT11 PICTURE XX                 PJJPLS
                   VALUE SPACE.                          PJJPLS
                   WR02-DAT12 PICTURE XX                 PJJPLS
                   VALUE SPACE.                          PJJPLS
                   WR02-DAT13 PICTURE XX                 PJJPLS
                   VALUE SPACE.                          PJJPLS
                   WR02-DAT113 PICTURE XX                PJJPLS
                   VALUE SPACE.                          PJJPLS
                   WR02-AMJ.                               PJJPLS
                   WR02-AMJA.                              PJJPLS
                   WR02-AMJA9 PICTURE 99                 PJJPLS
                   VALUE ZERO.                           PJJPLS
                   WR02-AMJM PICTURE XX                 PJJPLS
                   VALUE SPACE.                          PJJPLS
                   WR02-AMJJ PICTURE XX                 PJJPLS
                   VALUE SPACE.                          PJJPLS
                   WR02-BIS PICTURE 9                    PJJPLS
                   VALUE ZERO.                           PJJPLS
                   WR02-DHORDI.                           PJJPLS
                   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 '12/04/06'.	PJJPLS
	05	PROGR	PICTURE	X(6)	VALUE 'PJJPLS'.	PJJPLS
	05	CODUTI	PICTURE	X(6)	VALUE 'PTJJCQ'.	PJJPLS
	05	TIMGN	PICTURE	X(8)	VALUE '10:52:36'.	PJJPLS
	05	PROGE	PICTURE	X(8)	VALUE 'APPLICAT'.	PJJPLS
	05	COBASE	PICTURE	X(4)	VALUE 'HDEV'.	PJJPLS

05	DATGNC	PICTURE X(10)	VALUE	'12/04/2006'.	
05	RELEAS	PICTURE X(7)	VALUE	'3.5 V04'.	
05	DATGE	PICTURE X(10)	VALUE	'03/03/2006'.	
05	DATSQ	PICTURE X(10)	VALUE	'03/03/2006'.	
01	DATCE.				PJJPLS
05	CENTUR	PICTURE XX	VALUE	'20'.	PJJPLS
05	DATOR.				PJJPLS
10	DATOA	PICTURE XX.			PJJPLS
10	DATOM	PICTURE XX.			PJJPLS
10	DATOJ	PICTURE XX.			PJJPLS

Variables et indices

INDICATEURS DE FIN DE FICHER

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 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 RUB sont les valeurs fournies dans la zone complémentaire. Le test du FILE-STATUS n'est pas effectué par les opérateurs de traitement de fichier.

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

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.

Vous pouvez modifier le contenu de ces zones de travail par des lignes d'appel de Structures de Données. En aucun cas, vous ne pouvez 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, vous pouvez obtenir la description de la Structure de Données sous un niveau 01 ou dans une section (LINKAGE, COMMUNICATION, ...) que vous aurez 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.

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

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
F01LK-FN.	EXIT.	PJJPLS
N01TR.	NOTE *INITIALISATION FICHER TR-FICHER *.	PJJPLS
F01TR.	OPEN INPUT TR-FICHER.	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 FICHERS ACCES SEQ. SANS DE	PJJPLS
	* *	PJJPLS
	*****.	PJJPLS
N05LK.	NOTE *LECTURE FICHER LK SANS DE*.	PJJPLS

F05LK.	IF	LK-FT	= '0'		PJJPLS
	NEXT SENTENCE ELSE GO TO			F05LK-FN.	PJJPLS
F05LK-10.	RETURN	LK-FICHER	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 FICHER		TR SANS DE*.	PJJPLS
F05TR.	IF	TR-FT	= '0'		PJJPLS
	NEXT SENTENCE ELSE GO TO			F05TR-FN.	PJJPLS
F05TR-10.	READ	TR-FICHER	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-FICHER.			PJJPLS
F20KL-FN.	EXIT.				PJJPLS
F20LA.	CLOSE	LA-FICHER.			PJJPLS
F20LA-FN.	EXIT.				PJJPLS
F20TR.	CLOSE	TR-FICHER.			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 FICHER 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

Chapitre 9. Annexes

Source Cobol (Lignes '9')

Les lignes '9' sont du langage source COBOL.

Elles permettent :

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

GENERATION DE DONNEES

Il est possible de faire générer par VisualAge 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 VisualAge 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 VA Pac (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 VA Pac étant ignorées,
- si la variante correspond à un COBOL généré, le Programme sera généré, les lignes COBOL seront ignorées.

Si vous souhaitez voir apparaître dans votre Programme des renseignements identiques à ceux que vous trouverez dans tout Programme généré dans la zone CONSTANTES-PAC, il faut codifier une ligne 9 de la façon suivante :

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

Par exemple :

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

Le résultat donné est de la forme AAA9999V000 DDDDDDDDD, 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 vous souhaitez voir apparaître des informations plus complètes (code utilisateur, code de la Base, date de génération avec siècle...), vous devez coder une ligne 9 de la façon suivante :

```
cols 12                35  
01 CONSTANTES-PAC PICTURE X(87) 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.),

- Version (7 car.),
- Date du générateur (10 car.),
- Date du squelette (10 car.).

CODES OPERATION

C1 : valeur par défaut.

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

SUIVI PERMIS DE CONSTRUIRE
 TRAITEMENTS PROGRAMME LTVL1 FIRST TEST PROGRAM
 1

P0*DOC.LURE.JIA.893

```

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

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
1	6	CODE PROGRAMME
		Code identifiant le Programme dans la Bibliothèque.
2	1	CODE ACTION
	'C'	Création de la ligne
	'M'	Modification de la ligne
	'A' ou 'D'	Annulation de la ligne
	'T'	Transfert de la ligne
	'B'	Annulation multiple
	'G'	Transfert d'un groupe de lignes
	'?'	Demande de documentation
	'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
	'X'	Mise à jour implicite sans transformation majuscule/ minuscule
3	6	NUMERO DE LIGNE COBOL
		ZONE PSEUDO-NUMERIQUE

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			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.



Référence : DDSTR000352F - 7464

Imprimé en France