



VisualAge Pacbase 2.5

**VA PAC 2.5 – IBM DOS/VSE
MANUEL D'EXPLOITATION VOLUME I : ENVIRONNEMENT & INSTALLATION**

DEPDO001252F

Remarque

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

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

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

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

Deuxième Edition (Juin 1998)

La présente édition s'applique à :

- VisualAge Pacbase Version 2.5

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

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

ou en nous adressant un courrier à :

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

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

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

REMARQUES

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

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

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

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

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

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

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

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

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

MARQUES

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

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

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

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

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

TABLE DES MATIERES

1. PREAMBULE.....	7
2. LES COMPOSANTS DE VISUALAGE PACBASE.....	9
2.1. PRESENTATION GENERALE.....	10
2.2. CODIFICATION DES MODULES, EXTENSIONS, UTILITAIRES	11
2.3. BIBLIOTHEQUES DES PROGRAMMES TP.....	12
2.4. BIBLIOTHEQUES DES PROGRAMMES BATCH.....	16
2.5. BIBLIOTHEQUES DES PARAMETRES.....	21
2.6. BIBLIOTHEQUES DES PROCEDURES	22
2.7. LES PROCEDURES BATCH.....	23
2.8. LES FICHIERS SYSTEME	25
2.9. LES FICHIERS EVOLUTIFS.....	29
3. ENVIRONNEMENT.....	33
3.1. PRESENTATION GENERALE.....	34
3.2. ENVIRONNEMENT (CICS)	35
3.3. METHODES D'ACCES (VSAM).....	39
3.4. ENVIRONNEMENT BATCH.....	40
3.5. UTILISATION DES BANDES.....	41
3.6. ENCOMBREMENT DES FICHIERS.....	42
3.7. OPTIMISATION DES PERFORMANCES.....	45
3.8. INSTALLATION DES FICHIERS INDEX ET DONNEES	46
3.9. STRUCTURE DE VISUALAGE PACBASE SOUS CICS	47
3.10. RESSOURCES NECESSAIRES SOUS CICS.....	49
3.11. DETERMINATION DU NOMBRE DES REQUETES SOUS CICS.....	52
3.12. ANOMALIES D'EXECUTION	54
4. INSTALLATION.....	56
4.1. PREAMBULE.....	57
4.2. PRESENTATION GENERALE.....	59
4.3. BANDE D'IMPLANTATION.....	60
4.4. DECHARGEMENT DU JCL INITIAL	61
4.5. EXECUTION DU JCL INITIAL	63
4.6. INSTALLATION DU JCL COMPLET	68
4.7. VARIANTES D'INSTALLATION	70
4.8. LISTE DES MODULES DE JCL.....	73
4.9. PARAMETRAGE DU JCL.....	78
4.10. ECLATEMENT DES MODULES DE JCL	82
4.11. DEROULEMENT DE L'IMPLANTATION.....	85
4.12. 1. MODIFICATION DU JCL DE POWER	87
4.13. 2.1 MISE A JOUR DES TABLES CICS (CSD).....	89
4.14. 2.2 MISE A JOUR DES TABLES CICS (ASSEMBLEUR).....	95
4.15. 3.1 PROCEDURES SLI A CATALOGUER.....	103
4.16. 3.2 PROCEDURES VSE A CATALOGUER.....	111
4.17. 4.1 DEFINITION DES FICHIERS PERMANENTS	116
4.18. 4.2 PROCEDURE GPRT A CATALOGUER.....	119
4.19. 4.3 PROCEDURE PACX A CATALOGUER	123
4.20. 4.4 DEFINITION DES FICHIERS DE TRAVAIL DE GPRT	126
4.21. 4.5 DEFINITION DES FICHIERS DE TRAVAIL DE PACX.....	132
4.22. 4.6 INSTALLATION DES MODULES EXECUTABLES	135
4.23. 4.7 CHARGEMENT DES FICHIERS SYSTEME	142
4.24. 4.8 CHARGEMENT DES FICHIERS UTILISATEUR.....	145
4.25. 4.9 CHARGEMENT DES LIBELLES D'ERREURS.....	147
4.26. 5.1 CHARGEMENT BASE DE TESTS	149
4.27. 5.2 INIT COMMANDES D'EDITION-GENERATION.....	151
4.28. 5.3 INITIALISATION MODULE PEI.....	152

4.29.	6.1 COMPLEMENT: INSTALLATION DICTIONNAIRE VISUAL.....	153
4.30.	6.2.1 COMPLEMENT: ENVIRONNEMENT PAF.....	154
4.31.	6.2.2 COMPLEMENT: EXTENSION PAF+.....	155
4.32.	6.3 COMPLEMENT: MODULE STATION DE TRAVAIL.....	158
4.33.	6.4 COMPLEMENT: PACBENCH QUALITY CONTROL.....	159
4.34.	6.5 COMPLEMENT: INSTALLATION CARTES DE CONTROLE.....	160
4.35.	6.6 COMPLEMENT: CATALOGAGE PROCEDURE DCOB.....	164
4.36.	7. MODIFICATION DU JCL DE CICS.....	167
4.37.	8.1 COMPLEMENT: MODULE PAC/IMPACT.....	169
4.38.	8.2 COMPLEMENT: MODULE PAC/TRANSFERT.....	173
4.39.	8.3 COMPLEMENT: PONT VA PAC / TEAMCONNECTION.....	175
4.40.	9. LISTE DES PROGRAMMES ET FICHIERS INSTALLES.....	176
4.41.	TESTS D'INSTALLATION.....	178
4.42.	TESTS D'UTILISATION DE LA BASE.....	179
4.43.	TESTS DE GESTION DE LA BASE.....	181
4.44.	TESTS DES UTILITAIRES D'EXTRACTION.....	183
4.45.	TESTS D ' UTILISATION : PAC/IMPACT.....	185
4.46.	TESTS D ' UTILISATION : PAC/TRANSFERT.....	186
4.47.	TESTS D ' UTILISATION : PONT VA PAC/VA SMALLTALK.....	187
5.	REIMPLANTATION D'UNE SOUS-VERSION.....	188
5.1.	REIMPLANTATION STANDARD.....	189
6.	REPRISE VA PACBASE 8.0, ..., 1.6.....	191
6.1.	AVERTISSEMENT.....	192
6.2.	REPRISE VA PACBASE 8.0, 8.0.1.....	193
6.2.1.	OPERATIONS A EFFECTUER.....	193
6.2.2.	PE80 : REPRISE PARAMETRES UTILISATEUR.....	195
6.2.3.	PC80 : REPRISE DE LA BASE.....	196
6.2.4.	RPPG : REPRISE FICHER DES COMMANDES D'EDITION(AG).....	197
6.2.5.	PJ80 : REPRISE DE JOURNAL ARCHIVE.....	198
6.2.6.	PP16 : REPRISE ENVIRONNEMENT DE PRODUCTION.....	201
6.3.	REPRISE VA PACBASE 802.02, ..., 1.6.....	202
6.3.1.	OPERATIONS A EFFECTUER.....	202
6.3.2.	RPPG : REPRISE FICHER DES COMMANDES D'EDITION(AG).....	204
6.3.3.	PJ16 : REPRISE DU JOURNAL.....	205
6.3.4.	PP16 : REPRISE ENVIRONNEMENT DE PRODUCTION.....	206

1. PREAMBULE

AVERTISSEMENT

UTILISATION DU MANUEL ENVIRONNEMENT ET INSTALLATION

Ce manuel est destiné au responsable de l'installation. Il décrit les composants de VisualAge Pacbase et l'environnement système, les consignes d'installation de la nouvelle version et les opérations à effectuer pour une réimplantation standard de versions de correction.

UTILISATEURS DISPOSANT D'UNE VERSION ANTERIEURE

D'une manière générale, il est conseillé d'installer la nouvelle version dans un environnement entièrement distinct de toute version antérieure du système quant au choix des paramètres d'installation, et de procéder à des tests à l'aide du jeu d'essai fourni sur le support d'installation.

'Release' VisualAge Pacbase antérieure à 802.02

Pour traiter ces reprises, veuillez contacter le Support VisualAge Pacbase.

VisualAge Pacbase 802.02, 1.2, 1.5, 1.6, 2.0

Dans ce cas, il est possible de procéder à une installation dans le même environnement que la version ancienne. Se reporter, alors, au chapitre 'Reprise VisualAge Pacbase 802.02, 1.2, 1.5, 1.6' ou au chapitre 'Reprise VisualAge Pacbase 2.0'.

2. LES COMPOSANTS DE VISUALAGE PACBASE

LES COMPOSANTS DE VISUALAGE PACBASE	PAGE	10
PRESENTATION GENERALE		2
		1

2.1. PRESENTATION GENERALE

PRESENTATION GENERALE

Le système VisualAge Pacbase gère les données permanentes en mode conversationnel ou en mode batch grâce à l'utilisation de deux types de ressources :

- . Des BIBLIOTHEQUES dans lesquelles sont stockés les programmes constitutifs du système, ainsi que les paramètres nécessaires à son fonctionnement :

- Une bibliothèque de programmes TP,
- Une bibliothèque de programmes batch,
- Une bibliothèque de paramètres du système VisualAge Pacbase,
- Une bibliothèque de paramètres par Base VisualAge Pacbase,
- Une bibliothèque de JCL des procédures batch.

- . Des FICHIERS PERMANENTS qui matérialisent les données manipulées par les programmes définis précédemment. Ces fichiers peuvent être classés en deux catégories :
- . Les fichiers SYSTEME, qui ne sont pas liés à une Base VisualAge Pacbase particulière et restent relativement stables.
- . Les fichiers EVOLUTIFS qui correspondent à une Base VisualAge Pacbase et dont le volume varie en fonction des mises à jour effectuées.

REMARQUE

Les modules STATION DE TRAVAIL, DSMS, DSMS habillé, PAW, Pactables sont implantés indépendamment des autres modules de VisualAge Pacbase.

Le pont VisualAge Pacbase-ENDEVOR est également implanté indépendamment des autres modules.

Leur installation et leur exploitation sont décrites dans des Manuels d'exploitation spécifiques.

2.2. CODIFICATION DES MODULES, EXTENSIONS, UTILITAIRES

CODIFICATION DES MODULES, EXTENSIONS, UTILITAIRES

La codification pour les Modules, les Extensions et les Utilitaires optionnels est la suivante :

DICTIONNAIRE = DIC

.Extensions :

Gestionnaire de Documentation Personnalisée = PDM
Interface systèmes de sécurité = SEC

.Utilitaires optionnels :

Gestion de bases d'information réparties = LCU
Normalisation = RME
Analyse d'activité = ACT

MODULES :

LANGAGE STRUCTURE = SC
LANGAGE BATCH = BSD
GENERATEUR COBOL = COB
DIALOGUE = OSD
CLIENT/SERVEUR = OCS
DBD = DBD
DBD Relationnel SQL = SQL
PACTABLES = TAB
DSMS = DSM
ENVIRONNEMENT DE PRODUCTION = PEI
PERSONNALISATION = DEX
Pac/Transfer = TRF
Pont VA SMALLTALK / VA Pacbase = VIS
Pont VA Pacbase / TeamConnection = PTC
ANALYSE D'IMPACT = S2K
PACBENCH QUALITY CONTROL = PQC
STATION DE TRAVAIL = WST
PACBASE ACCESS FACILITY = PAF
PACREVERSE = REV
PACBASE WEB CONNECTION = PAW

NOTE : L'interface de sécurité et le module DB2 ne sont pas pris en compte dans la version CICS-DOS.

2.3. BIBLIOTHEQUES DES PROGRAMMES TP

LA BIBLIOTHEQUE DES PROGRAMMES TP

Suivant les Modules et Options installés, elle comprend les programmes suivants :

```

-----
! CODE      ! MODULE ! CHOIX CORRESPONDANT !
! PROGRAMME ! OPTION ! Commentaires        !
-----
!-----!-----! TRANSACTION xxEF    !
! xxPA00    ! DIC   ! .Ecran initial     !
! xxPA01    ! -    ! H                   !
! xxPA10    ! -    ! HP .                !
! xxPA11    ! -    ! LCPC..              !
! xxPA12    ! -    ! PC..                !
! xxPA13    ! -    ! PT.                 !
! xxPA14    ! -    ! PE.                 !
! xxPA15    ! -    ! PU.....            !
! xxPA16    ! -    ! PK                  !
! xxPA17    ! -    ! PD                  !
! xxPA18    ! -    ! LCPU.....          !
! xxPA19    ! -    ! PW                  !
! xxPA20    ! -    ! PB                  !
! xxPA21    ! -    ! PM                  !
! xxPA22    ! -    ! LCPM                !
! xxPA30    ! PEI   ! HE                  !
! xxPA31    ! -    ! EE.....            !
! xxPA32    ! -    ! EG.....            !
! xxPA33    ! -    ! ES.....            !
! xxPA34    ! -    ! LSEP.....          !
! xxPA35    ! -    ! ED.....            !
! xxPBNB    ! DIC   ! Map abend          !
! xxPPHP    ! -    ! Help                !
!-----!-----! TRANSACTION VA Pacbase !
! xxQA00    ! DIC   ! D..                 !
! xxQB00    ! BSD   ! R...                !
! xxQC00    ! DIC   ! E.....            !
! xxQC01    ! -    ! LUE                 !
! xxQC50    ! WST   ! ++5 Up/Dw maquettage !
! xxQD00    ! SC    ! P.....B et O.....B !
! xxQE00    ! DIC   ! E.....D            !
! xxQF00    ! COB   ! P.....SC           !
! xxQF10    ! -    ! P.....STR          !
! xxQG00    ! DIC   ! K.....             !
! xxQH00    ! OSD   ! O.....             !
! xxQH01    ! DIC   ! LA LC LE LF LM LN LP LS !
!          !      ! LT LX              !
! xxQH20    ! OSD   ! O.....CS           !
! xxQH30    ! -    ! O.....O            !
! xxQI00    ! -    ! O.....L            !
-----

```

LES COMPOSANTS DE VISUALAGE PACBASE
BIBLIOTHEQUES DES PROGRAMMES TP

2
3

! CODE	! MODULE	! CHOIX	! CORRESPONDANT	!
! PROGRAMME	! OPTION	! Commentaires		!
! xxQI01	! OSD	! O.....CE	(C1)	!
! xxQI02	! -	! O.....CE	(C2)	!
! xxQI03	! -	! O.....SIM		!
! xxQI04	! -	! O.....ADR		!
! xxQI05	! -	! O.....CE	(C3)	!
! xxQI20	! -	! O.....M		!
! xxQI21	! -	!		!
! xxQI50	! WST	! ++4 Up/Dw	rubr. écran	!
! xxQK10	! DIC	! M.....		!
! xxQK20	! -	! M.....CM		!
! xxQK30	! -	! M.....CE		!
! xxQL10	! -	! B.....		!
! xxQL20	! -	! B.....DH		!
! xxQL21	! -	! B.....DT		!
! xxQL30	! -	! B.....DC		!
! xxQL40	! SQL	! B.....DR...		!
! xxQL41	! -	! B.....DR...K		!
! xxQL45	! -	! B.....GEN		!
! xxQL46	! -	!		!
! xxQM00	! SC	! P.....CP	et O.....CP	!
! xxQP00	! -	! P.....P	et O.....P	!
! xxQP01	! -	! affichage	P.....TC	!
! xxQP02	! -	! affichage	O.....TC	!
! xxQP03	! -	! P.....TC	et O.....TC	!
! xxQP04	! -	! P.....TO		!
! xxQP05	! -	! O.....TO		!
! xxQP06	! -	! P.....PG	et O.....PG	!
! xxQP07	! -	! O.....PG		!
! xxQP08	! -	! P.....PG		!
! xxQP50	! WST	! ++6 Up/Dw	traitements	!
! xxQR00	! DIC	! LL.....L_.....		!
! xxQS02	! -	! -XP		!
! xxQS03	! -	! -ACT		!
! xxQS04	! -	! WS		!
! xxQS05	! -	! ?		!
! xxQS06	! -	! Menus		!
! xxQS08	! DEX	! -XQ		!
! xxQT00	! DIC	! T.....D		!
! xxQT10	! -	! T.....		!
! xxQT20	! PDM	! T.....SIM		!
! xxQT50	! WST	! ++2 Up/down	textes	!
!-----!	!-----!	! TRANSACTION	xxEF	!
! xxQUPA	! DIC	! Moniteur	anglais	!
! xxQUPF	! -	! Moniteur	français	!

```

-----
! CODE      ! MODULE ! CHOIX CORRESPONDANT !
! PROGRAMME ! OPTION ! Commentaires         !
-----
!-----!-----! TRANSACTION VA Pacbase !
! xxQU00    ! DIC    ! U..                  !
! xxQU01    ! -      ! U.....D            !
! xxQU10    ! PDM    ! V.....             !
! xxQU20    ! -      ! V.....D            !
! xxQV10    ! DIC    ! I.....             !
! xxQV20    ! -      ! I.....D            !
! xxQV30    ! -      ! -G                   !
! xxQX00    ! -      ! *                     !
! xxQX01    ! -      ! LH                    !
! xxQY01    ! DEX    ! F.....             !
! xxQY02    ! -      ! F.....CE           !
! xxQY03    ! -      ! $ .....            !
! xxQY04    ! -      ! $ .....D           !
! xxQY05    ! -      ! Q.....             !
! xxQY10    ! WST    ! ++1 Upload entités Design !
! xxQY11    ! -      ! ++3 Dwload - -      !
! xxQY20    ! DIC    ! GP                    !
! xxQY30    ! -      ! JO                     !
! xxQZ00    ! -      ! Ecran initial        !
! xxQ000    ! SC     ! P.....             !
! xxQ100    ! -      ! P.....CD           !
! xxQ101    ! -      ! P.....HCD          !
! xxQ102    ! -      !                       !
! xxQ103    ! -      !                       !
! xxQ104    ! -      !                       !
! xxQ200    ! DIC    ! S....               !
! xxQ210    ! TAB    ! S....SS             !
! xxQ300    ! DIC    ! S....CE             !
! xxQ400    ! BSD    ! R...L               !
! xxQ500    ! -      ! R...D               !
! xxQ600    ! -      ! R...CE              !
! xxQ700    ! SC     ! P.....W et O.....W !
! xxQ800    ! -      ! P.....8             !
! xxQ900    ! -      ! P.....9             !
! xxR000    ! DIC    ! Moniteur VA Pacbase !
! xxR005    ! -      ! Retour à CICS        !
! xxR100    ! -      ! Décodage zone CHOIX !
! xxR200    ! -      ! Conversion FORMATS   !
! xxR400    ! -      ! N*... ou NH....     !
! xxR500    ! -      ! Map abend            !
-----

```

```
-----!
! CODE      ! MODULE ! CHOIX CORRESPONDANT !
! PROGRAMME ! OPTION ! Commentaires        !
!-----!
!-----!-----! TRANSACTION VA Pacbase !
! xxR600    ! DIC   ! Traitement de texte  !
! xxR980    ! -    ! Formatage écran      !
! xxR990    ! -    ! Formatage pour le WEB !
! xxUCTA    ! -    ! Gestion UCTRAN (pgm ASM) !
!          !      ! jusqu'à CICS 2.1     !
! xxUCTR    ! -    ! Gestion UCTRAN (pgm COBOL) !
!          !      ! à partir de CICS 2.2  !
! xxUCTX    ! -    ! Neutralisation xxUCTR (pgm COBOL) !
!-----!
!-----!-----! Transaction PUF-TP !
! xxFANM    ! DIC   !                      !
! xxFBIB    ! -    !                      !
! xxFCTL    ! -    !                      !
! xxFDBD    ! -    !                      !
! xxFECD    ! -    !                      !
! xxFENU    ! -    !                      !
! xxFFOG    ! -    !                      !
! xxFMCL    ! -    !                      !
! xxFPGM    ! -    !                      !
! xxFRUB    ! -    !                      !
! xxFSDO    ! -    !                      !
! xxFTXT    ! -    !                      !
! xxFVER    ! -    !                      !
! xxF000    ! -    !                      !
!-----!
! Sous-programmes appelés dans les programmes générés : !
! xxTPST    ! PAF   ! PAF TP                !
! xxTPWS    ! -     ! PAF TP Mots-clés     !
! (à mettre en bibliothèque de programmes d'application) !
-----!
```

2.4. BIBLIOTHEQUES DES PROGRAMMES BATCH

LA BIBLIOTHEQUE DES PROGRAMMES BATCH

Suivant les Modules et Options installés, elle comprend les programmes suivants :

! Code	! Procédures	! Option	! Commentaires
! PACA05	! UPDT	! DIC	!
! PACA15	! UPDT REST	! -	!
! PACA10	! GPRT	! -	!
! PACA20	! -	! -	!
! PACB	! -	! -	! moniteur GPRT
! PACB30	! -	! -	!
! PACB31	! -	! SQL	!
! PACB40	! -	! DBD	!
! PACB80	! -	! -	!
! PACCTL	! PACX	! DIC	!
! PACC30	! GPRT	! COB	!
! PACC40	! -	! -	!
! PACC80	! -	! -	!
! PACDTP	! INSL	! DIC	!
! PACD30	! -	! -	!
! PACD40	! -	! -	!
! PACD80	! -	! -	!
! PACD90	! -	! -	!
! PACE30	! -	! OSD	!
! PACE40	! -	! -	!
! PACE80	! -	! -	!
! PACFGY	! PACX	! DIC	!
! PACFMB	! -	! -	!
! PACFTD	! -	! -	!
! PACG3C	! GPRT	! OCS	!
! PACG3S	! -	! -	!
! PACG4S	! -	! -	!
! PACG8C	! -	! -	!
! PACG8S	! -	! -	!
! PACHOI	! PACX	! DIC	!
! PACINS	! VINS	! -	!
! PACK30	! GPRT	! OCS	!
! PACK80	! -	! -	!
! PACK90	! -	! -	!
! PACL30	! -	! SC	!
! PACL80	! -	! -	!
! PACL90	! -	! -	!
! PACL92	! EMUP	! -	!
! PACL93	! EMLD	! -	!

! Code	! Procédures	! Option	! Commentaires
! PACL95	! GPRT	! PAW	!
! PACM30	! -	! DIC	!
! PACM80	! -	! -	!
! PACN30	! -	! PDM	!
! PACN35	! -	! -	!
! PACN40	! -	! -	!
! PACN50	! -	! -	!
! PACN80	! -	! -	!
! PACP30	! -	! SC	!
! PACP40	! -	! -	!
! PACP80	! -	! -	!
! PACP82	! -	! -	!
! PACQ	! PQCA	! PQC	! Moniteur PQCA
! PACR01	! INPE	! PEI	!
! PACR10	! PRPE	! -	!
! PACR20	! GPRT	! -	!
! PACR22	! SIPE	! PEI	!
! PACR30	! HIPE	! -	!
! PACR40	! GRPE	! -	!
! PACR60	! SVPE	! -	!
! PACR61	! RSPE	! -	!
! PACSJO	! PACX	! DIC	!
! PACSMD	! -	! -	!
! PACSPU	! -	! -	!
! PACSRM	! -	! -	!
! PACS30	! -	! -	!
! PACS40	! -	! -	!
! PACS50	! -	! -	!
! PACS60	! -	! -	!
! PACS75	! -	! -	!
! PACS80	! -	! -	!
! PACTIN	! GETI	! TAB	!
! PACTI1	! GET0	! -	!
! PACT40	! GETA GETD	! -	!
! PACT41	! GET1 GET2	! -	!
! PACT45	! GETA GETD	! -	!
! PACT50	! GETD	! -	!
! PACT51	! GET2	! -	!
! PACU15	! PARM	! DIC	!
! PACU80	! -	! -	!
! PACU99	! CRYP	! -	!
! PACX	! PACX	! -	!
! PADM10	! SADM	! WST	! Contrôle intégrité SSADM
! PAFP10	! PPAF GPRP	! PAF	! Pré-processeur PAF
! PAF900	! UPDP	! -	! Entrées m.à.j. PAF

! Code	! Procédures	! Option	! Commentaires
! PAN200	! INFQ	! S2K	!
! PAN205	! INFP	! -	!
! PAN210	! ISEP	! -	!
! PAN212	! ISOS	! -	!
! PAN215	! ISEP IANA	! -	!
! PAN220	! IPFQ IANA	! -	!
!	! IPEP	! -	!
! PAN230	! IANA	! -	!
! PAN240	! IPFQ	! -	!
! PAN250	! IANA	! -	!
! PAN255	! IGRA	! -	!
! PAN260	! IANA	! -	!
! PAN270	! IPIA	! -	!
! PAN280	! IPIA	! -	!
! PBBTST	! Sub-pgm	! PAF	! Appelé par pgms batch uti
! PBBTWS	! -	! -	! - - - - -
! PBBT98	! -	! -	! Appelé par PBBTST PBBTWS
! PDS600	! DEXP	! DSM	! DEXP < 2.0
! PDS610	! -	! -	! -
! PREI00	! RVDE	! REV	! Interface Pacreverse
! PREI40	! RVKE	! -	! -
! PREI50	! -	! -	! -
! PRE986	! RVDE	! -	! -
! PTARSD	! RMTD	! TAB	! Migration: rest. TD
! PTAR20	! RPTD	! -	! Migration: reprise TD
! PTASVD	! SMTD	! -	! Migration: sauvegarde TD
! PTC010	! TCLS	! PTC	! VA Pac-TeamConnection
! PTC030	! -	! -	! -
! PTC100	! TCGP	! -	! -
! PTC200	! TCME	! -	! -
! PTC220	! TCCI TCME	! -	! -
! PTC400	! TCCI	! -	! -
! PTC440	! -	! -	! -
! PTED30	! XPDM	! PDM	! Extension PDM
! PTED60	! -	! -	! -
! PTEP90	! PRGS	! -	! -
! PTEXD0	! XPAF	! PAF	! Extension PAF
! PTEX30	! -	! -	! -
! PTEX80	! -	! -	! -
! PTUBAS	! SAVE UPDT	! DIC	! Vérifie intégrité base
! PTUCSS	! CSES	! -	! Compression No session
! PTUESS	! ESES	! -	! Extraction No Session

! Code	! Procédures	! Option	! Commentaires
! PTUG05	! TRJC	! TRF	! Module Pac/Transfer
! PTUG06	! -	! -	! -
! PTUG07	! -	! -	! -
! PTUG10	! TRUP	! -	! -
! PTUG11	! -	! -	! -
! PTUG12	! -	! -	! -
! PTUG42	! TRDU	! -	! -
! PTUG44	! -	! -	! -
! PTUG46	! -	! -	! -
! PTUG50	! TRPF	! -	! -
! PTUG60	! TRRP	! -	! -
! PTUG61	! -	! -	! -
! PTUG90	! TRRT	! -	! -
! PTULOI	! RTLO	! DIC	! Reprise des verrous
! PTULVB	! LVBL	! -	! Remplacement low-values ! par des blancs
!	!	!	!
! PTUQ10	! PQCE	! PQC	!
! PTUQ15	! -	! -	!
! PTUQ20	! PQCA	! -	!
! PTUQ24	! -	! -	!
! PTUQ25	! -	! -	!
! PTUQ30	! -	! -	!
! PTUR00	! STOP	! -	!
! PTU001	!	! DIC	! Toutes procédures avec ! entrées utilisateur
!	!	!	!
! PTU004	! REST REAG	! -	! Contrôle code utilisateur!
! PTU100	! MLIB	! DIC	!
! PTU120	! -	! -	!
! PTU130	! SASN	! LCU	!
! PTU140	! -	! -	!
! PTU2CL	! REOR	! DIC	!
! PTU200	! -	! -	!
! PTU208	! -	! -	!
! PTU210	! -	! -	!
! PTU220	! -	! -	!
! PTU240	! -	! -	!
! PTU300	! ARCH	! DIC	!
! PTU320	! -	! -	!
! PTU380	! REST	! -	!
! PTU400	! -	! -	!
! PTU402	! RESY	! -	!
! PTU420	! REST	! -	!
! PTU500	! SAVE	! -	!
! PTU502	! SASY	! -	!

! Code	! Procédures	! Option	! Commentaires
! PTU550	! SVAG	! DIC	!
! PTU560	! REAG	! -	!
! PTU630	! ACTI	! ACT	!
! PTU640	! -	! -	!
! PTU810	! EMSN	! LCU	!
! PTU815	! MESN	! -	!
! PTU850	! CPSN	! -	!
! PTU855	! -	! -	!
! PVA100	! VDWN	! VIS	!
! PVA110	! -	! -	!
! PVA300	! VUP1	! -	!
! PVA310	! -	! -	!
! PVA320	! VUP2	! -	!
! PVA400	! VPUR	! -	!
! PYSMCC	! YSMC	! WST	! YSM consistency check
! PYSMC2	! -	! -	!
! PYSMC3	! -	! -	!
! UTIXSR	! UXSR	! DIC	!
! PACSECB	! SUB-PROG	! SEC	! Appelé par pgms VA Pac
! PACN25	! GPRT	! PDM	! - - -
! PACN90	! -	! -	! - - -
! PACSEP	! GPRT UPDT	! DIC	! Sous-pgm édition bannière
!	! REST	!	! de séparation des états
! PACA90	! GPRT UPDT	! -	! Sous-pgm d'analyse des
!	! REST	!	! formats des rubriques
! Reprises Pacbase 802.02, 1.2, 1.5, 1.6			
! PACR90	! PP16	! PEI	! Sauvegarde séquentielle
! REP2PJ	! PJ16	! DIC	! Reprise du journal
! Reprise Pacbase 2.0			
! PTU908	! RPPG	! DIC	! Comm. édition-génération

2.5. BIBLIOTHEQUES DES PARAMETRES

LA BIBLIOTHEQUE SOURCE DES MEMBRES 'SLI'

Sa taille n'excède pas une piste de 3380.

Elle contient des membres de JCL ou de commandes IDCAMS insérées par POWER dans les jobs au moment de l'exécution.\$ SLI').

Sous le nom DFxxyyff, se trouvent les commandes IDCAMS de définition des fichiers de la base, ff étant le code du fichier, xxyy étant un préfixe formé du radical du système et d'un code différent selon qu'il s'agit d'un fichier système ou utilisateur.

Sous le nom RGxxyyff, se trouvent les commandes IDCAMS de rotation des fichiers de sauvegarde, permettant de gérer de façon centralisée, le nombre de générations conservées pour celles-ci.

Sous le nom DBJBxxyy, se trouve le JCL inséré en début de chaque job (définition de ressources, normalisation).

Enfin, sous le nom ppppxxyy se trouvent les JCLs de certaines procédures, pppp étant le code de la procédure.

Afin de simplifier l'installation, il est souhaitable d'utiliser une bibliothèque existante déjà déclarée dans la LIBDEF SOURCE,SEARCH du JCL de POWER.

2.6. BIBLIOTHEQUES DES PROCEDURES

LA BIBLIOTHEQUE DES PROCEDURES VSE :

Sa taille n'excède pas une piste de 3380.

Elle contient essentiellement les procédures de JCL paramétrées, de définition des espaces de tri ou de travail non-VSAM.

Elle doit être commune aux systèmes VA PAC , PACTABLE et DSMS, afin d'éviter les risques de concurrence entre les espaces de travail de plusieurs partitions.

2.7. LES PROCEDURES BATCH

LES PROCEDURES BATCH

Les traitements BATCH sont regroupés en procédures qui sont décrites dans les Parties II et III des Manuels d'Exploitation ("Procédures Batch: Guide de l'administrateur" et "Procédures Batch: Guide de l'utilisateur").

Pour chaque procédure, on trouvera :

- . Une description générale comprenant :
 - une présentation,
 - les conditions d'exécution,
 - les actions à entreprendre en cas d'anomalie d'exécution.
- . La description des entrées utilisateur, des traitements et des résultats obtenus, ainsi que les recommandations éventuelles pour l'utilisation.
- . La description des étapes :
 - liste des fichiers utilisés (intermédiaires et permanents),
 - codes retour éventuels émis par chaque étape.
- . Le fichier de commandes.

CLASSEMENT DES PROCEDURES

Les procédures batch sont documentées dans les manuels suivants:

"Procédures Batch : Guide de l'administrateur".

- 1) Procédures de gestion de la base.
- 2) Administration des versions (PEI et Pac/transfer).
- 3) Utilitaires du gestionnaire.
- 4) Migrations.

"Procédures Batch : Guide de l'utilisateur".

- 1) Procédures standard.
- 2) Extraction personnalisée et documentation auto.
- 3) Suivi et contrôle qualité.
- 4) Méthodologie contrôle d'intégrité.
- 5) Pactables.
- 6) Analyse d'impact.
- 7) Pont VisualAge Smalltalk / VisualAge Pacbase.

Environnement et installation.

Procédures de reprise de versions antérieures :

. Versions 802.02 à 1.6 :

- reprise du journal archive (PJ16)
- reprise sauvegarde séquentielle PEI (PP16)

. Versions 2.0 :

- reprise du fichier des commandes d'édition-génération (RPPG)

2.8. LES FICHIERS SYSTEME

LES FICHIERS SYSTEME

En plus des bibliothèques décrites dans les sous-chapitres précédents, le système VA PAC comprend les fichiers permanents suivants :

. Un fichier contenant les LIBELLES D'ERREURS et la DOCUMENTATION EN LIGNE du système VA PAC : AE0.

- . Taille : Environ 32000 enregistrements
- . Organisation : SAM-ESDS
- . Recsize : 80
- . Utilisation : Batch
- . Définition : Membre SLI DF\$PRO.\$PRO.AE0

Ce fichier n'est pas directement utilisé par le fonctionnement courant du système. Il est fusionné avec les PARAMETRES UTILISATEUR nécessaires au fonctionnement TP du système (*) pour constituer le fichier système : AE .

- . Taille : = AE0 + paramètres utilisateur
- . Organisation : VSAM-KSDS
- . Recsize : 80
- . Clé : 12 (position 0)
- . Utilisation : Batch et TP
- . Définition : Membre SLI DF\$PRO.\$PRO.AE

. Un fichier contenant les PARAMETRES UTILISATEUR nécessaires au seul fonctionnement batch du système (*) : AP.

- . Taille : paramètres utilisateur
- . Organisation : VSAM-KSDS
- . Recsize : 80
- . Clé : 7 (position 0)
- . Utilisation : Batch et TP
- . Définition : Membre SLI DF\$PRO.\$PRO.AP

(*) Les PARAMETRES UTILISATEUR gérés dans le fichier AE sont : les codes utilisateur, les "clés d'accès" VA PAC , l'activation de l'Interface système de sécurité, l'activation du contrôle des mots de passe à blanc, l'activation du contrôle d'exécution des procédures batch, les types de textes, la gestion de la conversion de caractères accentués, le paramétrage des méthodes Pacdesign, la connexion Dsms-VA PAC .

Les PARAMETRES UTILISATEUR gérés dans le fichier AP sont : les modifications des parties fixes des libellés d'erreurs standard et les cartes de contrôle nécessaires à la génération des programmes.

Tous les paramètres utilisateur sont gérés par une transaction et une procédure batch particulières (cf. chapitre MAJ DES PARAMETRES UTILISATEURS du MANUEL D'EXPLOITATION, Partie II, LES PROCEDURES BATCH). C'est cette procédure de mise à jour PARM qui constitue les fichiers AE et AP et gère également :

. Un fichier sauvegarde des PARAMETRES UTILISATEUR : PE.

. Organisation : Séquentielle (SAM-ESDS ou SAM)
. Nom physique : \$PXU.SEQ.\$PRO\$PRO.PE
. Format : Fixe
. Longueur : 80
. Utilisation : Batch
. Rotation : Membre SLI RG\$PRO.\$PRO.PE

Ce fichier constitue la sauvegarde de TOUS LES PARAMETRES UTILISATEUR se trouvant dans les fichiers AE et AP.

. Un fichier squelette de génération SC, utilisé par les modules LANGAGE STRUCTURE et LANGAGE BATCH.

- . Taille : Environ 40 enregistrements
- . Organisation : VSAM-KSDS
- . Recsize : 3204
- . Clé : 4 (position 0)
- . Définition : Membre SLI DF\$PRO.\$PRO.SC

. Un fichier squelette de génération SG, utilisé par les modules DIALOGUE, DBD et DICTIONNAIRE.

- . Taille : Environ 360 enregistrements
- . Organisation : VSAM-KSDS
- . Recsize : 4605
- . Clé : 5 (position 0)
- . Utilisation : Batch uniquement
- . Définition : Membre SLI DF\$PRO.\$PRO.SG

. Un fichier squelette de génération SR, utilisé par le module GENERATEUR COBOL.

- . Taille : Environ 25 enregistrements
- . Organisation : VSAM-KSDS
- . Recsize : 4605
- . Clé : 5 (position 0)
- . Utilisation : Batch uniquement
- . Définition : Membre SLI DF\$PRO.\$PRO.SR

. Un fichier squelette de génération SP, utilisé par la génération d'extracteurs de l'EXTENSION AU MODULE PAF.

- . Taille : Environ 5 enregistrements
- . Organisation : VSAM-KSDS
- . Recsize : 4605
- . Clé : 5 (position 0)
- . Utilisation : Batch uniquement
- . Définition : Membre SLI DF\$PRO.\$PRO.SP

. Un fichier squelette SF, utilisé par la génération d'extracteurs de l'EXTENSION AU MODULE PAF.

- . Taille : Environ 3000 enregistrements
- . Organisation : SAM-ESDS
- . Recsize : 119
- . Utilisation : Batch uniquement
- . Définition : Membre SLI DF\$PRO.\$PRO.SF

. Un fichier squelette de génération SS, utilisé par le module DIALOGUE CLIENT/SERVEUR.

- . Taille : Environ 300 enregistrements
- . Organisation : VSAM-KSDS
- . Recsize : 4605
- . Clé : 5 (position 0)
- . Utilisation : Batch uniquement
- . Définition : Membre SLI DF\$PRO.\$PRO.SS

MODULE PAF (VA PAC ACCESS FACILITY)

Un fichier indexé de travail est nécessaire à l'utilisation du module PAF pour l'ensemble des programmes TP utilisateur accédant aux bases de même radical.

. Fichier de travail PAF TP (PA)

```
. Organisation : VSAM-KSDS
. Format       : variable
. Recsize     : moyen 200, maxi 539
. Clé        : 37 (position 2)
. Utilisation : Mis à jour par xxTPST et xxTPWS,
                sous-programmes appelés par les
                programmes TP utilisateur
. Définition  : Membre SLI DF$PRO.$PRO.PA
```

Un fichier indexé de travail est nécessaire à l'utilisation du module PAF par les programmes batch utilisateur. Ce fichier est défini avec l'option NOALLOCATION et peut être, au choix, détruit en fin de job ou simplement désalloué (au moyen d'un IDCAMS VERIFY avec DISP=NEW dans la DLBL).

S'il est envisagé d'exécuter simultanément des programmes PAF utilisateur dans plusieurs partitions, le caractère '%' doit être utilisé en première position du nom physique. Ceci génère automatiquement un suffixe lié à la partition, concaténé au nom du fichier.

. Fichier de travail PAF batch

```
. Organisation : VSAM-KSDS
. Format       : variable
. Recsize     : moyen 170, maxi 468
. Clé        : 12 (position 0)
. Utilisation : Mis à jour par le sous-programme
                PBBT98 appelé par PBBTST/PBBTWS
                (sous-programmes PAF appelés par
                les programmes utilisateur)
. Définition  : membre SLI DFSYSPAF
```

EXTENSION MODULE PAF

. Un fichier Schémas d'Extraction GS contenant les extracteurs de l'utilisateur et ses macro-commandes.

```
. Organisation : VSAM-KSDS
. Recsize     : 203
. Clé        : 25 (position 0)
. Utilisation : Batch et TP
. Définition  : Membre SLI DF$PRO.$PRO.GS
```

2.9. LES FICHIERS EVOLUTIFS

LES FICHIERS DE LA BASE VA PAC

Les quatre premiers constituent la base VA PAC proprement dite. Ils contiennent toutes les données relatives au développement des applications.

. Le fichier des données AR

- . Organisation : VSAM-RRDS
- . Recsize : 140
- . Utilisation : Batch et TP
- . Définition : Membre SLI DF\$PRO.\$PFI.AR

Toute ligne VA PAC est stockée dans le fichier des données sous un numéro interne invariant. Les états successifs d'une même ligne, concernant des sessions archivées différentes forment une chaîne dont la tête est l'état de la ligne le plus récent, et la fin l'état de la ligne le plus ancien. Les programmes n'accèdent jamais à une ligne VA PAC directement par ce fichier, mais obtiennent préalablement le numéro de la tête de chaîne par consultation du fichier des index (AN).

. Le fichier des index AN.

- . Organisation : VSAM-KSDS
- . Recsize : 54
- . Clé : 43 (position 0)
- . Utilisation : Batch et TP
- . Définition : Membre SLI DF\$PRO.\$PFI.AN

Le contenu de la clé du fichier index décrit les visions du fichier VA PAC offertes à l'utilisateur. Il identifie la ligne VA PAC en fonction de sa position dans le réseau et de l'angle de la consultation, et assure des fonctions techniques annexes.

L'information principale qui y est portée est le numéro interne de la ligne VA PAC sur laquelle pointe l'index.

. Le fichier des commandes d'édition-génération AG.

```
. Organisation : VSAM-KSDS
. Reccsize    : 150
. Clé        : 26 (position 0)
. Utilisation : Batch et TP
. Définition  : Membre SLI DF$PRO.$PFI.AG
```

C'est la zone de stockage utilisateur qui permet de saisir et de manipuler les commandes d'édition-génération. Ce fichier est peu volumineux, mais subit un fort taux de mise à jour en fonctionnement courant. Il est sauvegardé par la procédure SVAG, initialisé, restauré et éventuellement réorganisé par la procédure REAG.

. Le fichier journal AJ.

```
. Organisation : VSAM-RRDS
. Reccsize    : 167
. Utilisation : Batch et TP
. Définition  : Membre SLI DF$PRO.$PFI.AJ
```

Tous les mouvements passés sur la base, que ce soit en batch batch ou en TP, sont conservés afin de permettre une restauration de la base en cas de défaillance des sécurités standard du système, et d'autre part de fournir la source d'informations à l'outil d'analyse de l'évolution de la base.

Ces mouvements sont normalement stockés dans le fichier de sauvegarde des mouvements (PJ), le fichier des mouvements ne servant que de transit entre le moment où le système les traite et le moment où la procédure ARCH les conduit sur leur support définitif.

REMARQUE

Les mouvements passés sur le fichier des commandes d'édition génération (AG) ne sont pas conservés dans le fichier journal (AJ).

La sauvegarde du réseau est constituée de trois fichiers séquentiels qui peuvent se trouver soit sur disque, soit sur sur bande.

Lorsqu'elles sont sur disque, le contrôle du nombre de générations conservées à chaque instant est centralisé dans un membre SLI. A l'installation, les générations conservées sont G(-1) et G(0) et codifiées OLDff et ff.

. La sauvegarde de la base (PC)

```
. Organisation : Séquentielle (SAM-ESDS ou SAM)
. Nom physique : $PXU..SEQ.$PRO$PFI.PC
. Format       : Variable
. Utilisation  : Batch
. Encombrement : 149 octets par donnée
               55 octets par index
. Définition  : Membre SLI DF$PRO.$PFI.PC
. Rotation    : Membre SLI RG$PRO.$PFI.PC
```

C'est un fichier banalisé de sauvegarde sous forme séquentielle des constituants de la base VA PAC : index (AN), données (AR).

. La sauvegarde du journal (PJ)

. Organisation : Séquentielle (SAM-ESDS ou SAM)
. Nom physique : \$PXU..SEQ.\$PRO\$PFI.PJ
. Format : Fixe
. Longueur : 167
. Utilisation : Batch
. Rotation : Membre SLI RG\$PRO.\$PFI.PJ

Son objectif est d'accumuler tous les mouvements de mise à jour de la base VA PAC depuis son installation qui ont transité par le fichier des mouvements (AJ).

Au cas où sa taille deviendrait incompatible avec les contraintes d'exploitation, la procédure ARCH permet de le scinder en plusieurs fichiers dont seul le plus récent est manipulé régulièrement.

. La sauvegarde des commandes d'édition-génération (PG)

. Organisation : Séquentielle (SAM-ESDS ou SAM)
. Nom physique : \$PXU..SEQ.\$PRO\$PFI.PG
. Format : Fixe
. Longueur : 150
. Utilisation : Batch
. Rotation : Membre SLI RG\$PRO.\$PFI.PG

Son objectif est de sauvegarder les commandes d'édition-génération et également d'appliquer à ces commandes la réorganisation, par la procédure REAG.

MODULE ENVIRONNEMENT DE PRODUCTION (PEI)

Pour les utilisateurs disposant du Module PEI (voir aussi le chapitre correspondant à ce Module), trois fichiers supplémentaires sont gérés en batch et en TP. Ils contiennent les données nécessaires à la gestion de la mise en production des entités:

. Le fichier environnement de production en batch AB.

```
. Organisation : VSAM-KSDS
. Recsize      : 110
. Clé          : 26 (position 0)
. Utilisation  : Batch et consulté en TP
. Définition   : Membre SLI DF$PRO.$PFI.AB
```

. Le fichier environnement de production en TP AC.

```
. Organisation : VSAM-KSDS
. Recsize      : 110
. Clé          : 26 (position 0)
. Utilisation  : Batch et TP
. Définition   : Membre SLI DF$PRO.$PFI.AC
```

. La sauvegarde des fichiers de gestion des environnements de production PP.

```
. Organisation : Séquentielle (SAM-ESDS ou SAM)
. Nom physique : $PXU..SEQ.$PRO$PFI.PP
. Format        : Fixe
. Longueur     : 110
. Utilisation  : Batch
. Rotation     : Membre SLI RG$PRO.$PFI.PP
```

MODULE DSMS (DSM)

Pour les utilisateurs disposant du Module DSMS (voir également le Manuel d'exploitation de ce Module), un fichier de ce Module est consulté en batch et en TP par VA PAC .

Ce fichier contient la liste des entités concernées par chaque AMELIORATION. Le numéro d'AMELIORATION est renseigné par l'utilisateur lors de sa connexion à la base.

. Le fichier DSMS des éléments VA PAC DC.

```
. Organisation : VSAM-KSDS
. Format        : Variable
. Recsize      : mini 50, maxi 168
. Clé          : 31 (position 2)
. Utilisation  : Consulté par les maj TP et batch
```

L'allocation et l'initialisation de ce fichier sont effectuées lors de l'installation du Module DSMS.

3. ENVIRONNEMENT

3.1. PRESENTATION GENERALE

PRESENTATION GENERALE

Ce chapitre définit l'environnement et les ressources nécessaires au fonctionnement de VisualAge Pacbase et permet de prévoir les volumes "Disque" qui devront être disponibles.

3.2. ENVIRONNEMENT (CICS)

ENVIRONNEMENT CICS

ENVIRONNEMENT TP

Le moniteur utilisé est CICS VSE à partir de la Release 1.7.

Il doit comporter BMS (l'option PAGING n'est pas utilisée par le système VA PACBASE) avec l'option MAPALGN=NO.

Il doit avoir l'option SPOOL=YES, pour le lancement de jobs batch (Choix GP).

Les fichiers Données (AR), Index (AN), Journal (AJ), et demandes d'édition et de génération (AG), ainsi que le fichier des libellés d'erreurs (AE) et le fichier des paramètres utilisateur (AP), sont mis à jour en TP : ils doivent être protégés par l'option permettant le "Dynamic Backout".

Il en est de même pour le fichier TP du Module Environnement de production (AC).

Le nombre maximum de strings utilisés par fichier est de 2. Ce nombre peut donc être ajusté sans restriction selon la place disponible dans la partition de CICS (voir sous-chapitre "Optimisation des performances").

Il est conseillé d'utiliser l'option 'EMERGENCY RESTART' de CICS.

La taille de la COMMAREA utilisée est de 3,5 K.

La taille moyenne des programmes TP est de 48K, le plus important étant de 115 K.

CARACTERISTIQUES GENERALES DU FONCTIONNEMENT DU
SYSTEME

Les caractéristiques générales du fonctionnement du système sont les suivantes :

TRANSACTION VA PACBASE

- Un seul code transaction est utilisé pour la gestion de la base VA PACBASE. Il pointe sur un programme "moniteur" dont le but est d'enchaîner les différents programmes constitutifs du système. Ce programme ne gère directement aucun écran. (Code du programme : R000).
- . A chaque écran permettant la mise à jour est associé un programme (exemple : QC00 assure la mise à jour de la définition d'une Rubrique).
- . Les écrans qui affichent des listes "simples" sont traités dans un seul programme (QH01).
- . Les listes dites "spéciales", comme les références croisées, l'affichage des mots-clés, etc., sont traitées par des programmes spécifiques. (QS02, QS03, QS04 et QS08).
- . Les menus sont traités par le programme QS06.
- . Certains programmes sont appelés par la commande LINK. C'est en particulier le cas du programme de décodage de la zone choix (R100), et de celui qui contrôle les formats des rubriques (R200).
- . En cas d'anomalie gérée par le système, celui-ci renvoie une 'MAP ABEND'. Le programme d'affichage de cette map (R500) est appelé par XCTL.
- . Les mises à jour sont sérialisées, c'est-à-dire que le système empêche les accès concurrents au réseau par mise en file d'attente des transactions de mise à jour (ENQUEUE et DEQUEUE dans le cycle de mise à jour).

- Un programme particulier est systématiquement appelé lorsque l'utilisateur sort de la transaction VA PACBASE (Code du programme : R005). Ce programme effectue un EXEC CICS RETURN, après avoir éventuellement affiché 'LA CONVERSATION EN COURS A ETE SAUVEGARDEE' dans le cas d'une sortie par le choix '12', ou par la touche fonction correspondante (PF12 par défaut).

Si l'utilisateur désire que le système VA PACBASE ne rende pas la main à CICS, il peut remplacer ce programme par un programme utilisateur contenant par exemple l'appel d'une transaction utilisateur.

VA PACBASE envoie à ce programme une COMMAREA de trois caractères :

Position 1-2 : blanc ou 12 (pour sauvegarde de la conversation)
Position 3 : 'F' ou 'A' en fonction de la langue de l'installation.

Si l'utilisateur remplace le R005 par un programme qui lui est propre, il peut récupérer la COMMAREA et, s'il trouve '12' en positions 1-2, afficher le message 'LA CONVERSATION EN COURS A ETE SAUVEGARDEE' (si 'F' en position 3) ou 'CURRENT PACBASE CONVERSATION IS SAVED' (si 'A' en position 3).

TRANSACTION SPECIALE

- Une transaction particulière est utilisée pour :
 - . la gestion des paramètres utilisateur (en standard),
 - . la gestion de l'environnement de production (pour les utilisateurs disposant du Module PEI).

TRAITEMENT DES MAJUSCULES ET MINUSCULES

Le système VA PACBASE possède sa propre gestion des caractères en entrée :

- . Tous les codes introduits en minuscules sont transformés automatiquement en majuscules.
- . Tous les noms en clair des entités, ainsi que les textes sont laissés en minuscules. Les mots-clés implicites construits à partir des noms en clair sont transformés en majuscules.

Le code action 'X' permet de ne pas effectuer de transformation des caractères.

L'option UCTRAN du terminal, si elle est active, est INHIBEE AUTOMATIQUEMENT par PACBASE pour la durée de la connexion, ce qui permet de bénéficier de cette gestion.

Il est cependant possible de travailler en majuscules, en neutralisant la routine standard fournie.

Pour cela, il faut renommer sous un autre nom le programme xxUCTR, puis renommer le programme xxUCTX en xxUCTR.

Pour un CICS antérieur à 2.2, il faut renommer le programme xxUCTR, puis renommer le programme xxUCTA en xxUCTR.
Ce programme étant en assembleur, il faut ajouter LANGUAGE(ASSEMBLER) dans la ligne de définition du programme xxUCTR dans le membre 1CSD dans le cas du RDO et supprimer PGMLANG=COBOL dans le membre 1PPT sinon.

EN BATCH, la transcodification des minuscules est effectuée suivant les mêmes règles qu'en TP.

Si l'utilisateur veut travailler en majuscules seulement, il faut utiliser les fonctionnalités de l'éditeur contenant les mouvements batch Pacbase pour assurer la transformation automatique en majuscules, Pacbase n'assurant pas en batch cette transformation.

EN BATCH, la transcodification des minuscules est effectuée suivant les mêmes règles qu'en TP.

Si l'utilisateur veut travailler en majuscules seulement, il faut utiliser les fonctionnalités de l'éditeur contenant les mouvements batch pour assurer la transformation automatique en majuscules, VA Pacbase n'assurant pas en batch cette transformation.

3.3. METHODES D'ACCES (VSAM)

METHODES D'ACCES

Le système VisualAge Pacbase gère ses fichiers à l'aide des méthodes d'accès indexée VSAM-KSDS sans index secondaire et relative VSAM-RRDS.

Tous les fichiers sont protégés contre les accès concurrents en écriture (SHARE OPTION 2). En particulier, le système VA Pac se protège contre les mises à jour simultanées en batch et en TP uniquement par cette option de partageabilité.

Module Environnement de production :

Le fichier AB peut être mis à jour en batch simultanément par plusieurs exécutions de la procédure GPRT et comporte pour cela SHARE OPTION 4.

Toutes les procédures batch prévoient des DELETE/DEFINE en cas de rechargement des fichiers ; ceux-ci n'ont donc pas l'obligation d'avoir l'option REUSE fournie par défaut.

3.4. ENVIRONNEMENT BATCH

FICHIERS TEMPORAIRES

En mode batch, le fonctionnement du système utilise des fonctions standards du système d'exploitation, et la méthode d'accès VSAM.

Les fichiers de travail internes aux procédures sont essentiellement de type SAM-ESDS. Ils sont définis et détruits à chaque utilisation, exceptés les fichiers internes aux procédures GPRT et PACX, dont l'enveloppe, c'est-à-dire l'entrée dans le catalogue VSAM, est permanente. En revanche, ils occupent une place nulle lorsqu'ils ne sont pas en cours d'utilisation.

TAILLE DE LA PARTITION

La taille mémoire nécessaire à l'exécution des procédures batch varie essentiellement en fonction de la taille des buffers alloués aux fichiers qu'elles utilisent.

Il faut distinguer trois paramètres :

- La taille donnée dans le paramètre 'SIZE' de la carte EXEC qui concerne la zone de chargement des modules exécutables
- La taille de la GETVIS AREA, utilisée notamment par VSAM pour ses buffers,
- La taille totale de la partition, définie pour le système, qui totalise les deux aires précédentes.

Les informations ci-dessous concernent les seuls programmes ne pouvant pas s'exécuter dans une partition de 512K.

! PROC	! PROGRAMME	! SIZE	! GETVIS	! COMMENTAIRE	!
! REOR	! PTU220	! 870K	! 250K	!	!
! GPRT	! PACE80	! 425K	! 350K	! Générateur dialogue	!
!	! PAFP10	! 520K	! 50K	! Précompilation prog.	!
!	!	!	!	! utilisateur PAF	!
! SASN	! PTU140	! 560K	! 150K	!	!
! 9INS	! PACDTP	! 910K	! 150K	!	!

3.5. UTILISATION DES BANDES

UTILISATION DES BANDES

Lorsque des fichiers peuvent être volumineux, il est prévu de les diriger sur bande.

L'unité logique (SYSnnn) associée au fichier concerné et qu'il faudra assigner à un dérouleur est indiquée par un commentaire dans les procédures.

Le facteur de blocage maximum des fichiers sur bande est toujours permis ; la lecture de fichiers ayant un facteur de blocage plus petit est donc transparente.

3.6. ENCOMBREMENT DES FICHIERS

ENCOMBREMENT DES FICHIERS

L'encombrement total des fichiers dépend de l'importance des applications gérées par le système.

Le tableau ci-dessous indique l'espace disque nécessaire à une IMPLANTATION STANDARD :

! Encombrement	! Approximation			!	
! installation	! en millions d'octets			!	
! VisualAge Pacbase	! TOTAL	! dont	! VSAM	! NON VSAM	!
! TOTAL INSTALLATION	55	24	31	!	
! Total Système VA Pac	35	7	28	!	
! Total fichiers Utilisateur	20	17	3	!	
! de tests d'installation				!	

L'encombrement des fichiers système VA Pac reste fixe.

L'encombrement des fichiers utilisateur est expliqué à la page suivante. Une approximation permettant de prévoir globalement l'espace à allouer est de la calculer en comptant 400 octets par donnée VA Pac, tous fichiers confondus.

L'ENCOMBREMENT DES FICHIERS UTILISATEUR, après implantation, est à calculer en fonction des remarques suivantes.

Soit NPAC le nombre d'enregistrements VA Pac, toutes bibliothèques et sessions confondues. On a alors :

- (AR) Fichier Données : NPAC enregistrements de 140 octets.
- (AN) Fichier Index : Environ 3 * NPAC enregistrements de 54 octets (une donnée est en moyenne utilisée 3 fois) auxquels il faut ajouter l'espace libre (FSPC) introduit à la définition du fichier.
- (AG) Fichier des demandes d'édition - génération : Son volume est généralement faible. On peut estimer qu'il doit pouvoir contenir une centaine de demandes par utilisateur pouvant accéder au système (150 octets).
- (AJ) Fichier Journal : Il doit pouvoir contenir tous les mouvements de mise à jour, en batch ou en TP, passés entre 2 réinitialisations de ce fichier journal. Un mouvement VA Pac correspond à un enregistrement de 167 octets dans le fichier Journal.
- (AP) Fichier de Paramètres Utilisateurs : son volume est faible. Il contient les cartes de JCL du site ainsi que les éventuelles modifications de libellés d'erreurs standard (80 octets/enregistrement).

CALCUL DE L'ENCOMBREMENT DES FICHIERS VSAM

Fichier KSDS :
Nb enreg./CI = (CISIZE DATA - 10 - Free space CI) / RECSZ

Fichier RRDS :
Nb enreg./CI = (CISIZE - 4) / (RECSZ + 3)

EXEMPLE POUR UNE BASE CONTENANT 1 000 000 DE DONNEES

Fichier AN : DATA CISIZE(4096) RECSZ(54) FSPC(10,6)

- . Free space CI = 10 % de 4096 = 409
- . Nb. enreg. par CI = (4096 - 10 - 409) / 54 = 71
- . Nb. de CI par piste disque 3380 = 10
(Voir 'VSAM Administration Guide' IBM GC26-4066-1)
- . Nb. maxi CI/CA (1 CA = 1 Cylindre 3380) = 10 * 15 = 150
- . Free space CA = 6 % de 150 = 9
- . Nb de CI utilisés par CA = 150 - 9 = 141

Pour 3 000 000 d'enregistrements (3 * NPAC), on a donc en chargement environ :

$$3\ 000\ 000 / 71 = 42254\ \text{CI}$$
$$42254 / 141 = 300\ \text{CA DATA}$$

soit environ 300 cylindres de disque 3380.

Fichier AR : CISIZE(2048) RECSZ(140)

- . Nb. enreg. par CI = (2048 - 4) / (140 + 3) = 14
- . Nb. de CI par piste disque 3380 = 18
(Voir 'VSAM Administration Guide' IBM GC26-4066-1)
- . Nb. CI/CA (1 CA = 1 Cylindre 3380) = 18 * 15 = 270

Pour 1 000 000 d'enregistrements (NPAC), on a donc environ

$$1\ 000\ 000 / 14 = 71429\ \text{CI}$$
$$71429 / 270 = 265\ \text{CA}$$

soit environ 265 cylindres de disque 3380.

Soit un total d'environ 400 millions d'octets pour les deux fichiers évolutifs les plus volumineux du réseau.

MODULE ENVIRONNEMENT DE PRODUCTION (PEI)

Les fichiers AB et AC contiennent les mêmes informations ; la taille allouée à ces deux fichiers doit donc être la même, et se calcule selon la règle suivante :

Soit :

- . La longueur d'enregistrement = 110
- . Le nombre de bibliothèques de load-modules = NLM
- . Le nombre d'entités en production = NEP
- . Le nombre d'entités générées par bibliothèque de load modules = NEG

La taille allouée doit être en octets de :

$$110 * (\text{NLM} + (2 * \text{NEP}) + \text{NEG})$$

3.7. OPTIMISATION DES PERFORMANCES

OPTIMISATION DES PERFORMANCES

Plusieurs moyens permettent d'améliorer les performances de VA Pac en batch et en TP, en fonction de l'environnement sur lequel le système est installé et de l'importance des bases VA Pac gérées.

Ce chapitre fournit les informations nécessaires pour l'optimisation des performances.

Les options par défaut d'installation des fichiers, programmes et JCL peuvent être modifiées suivant les suggestions données ci-après.

Des optimisations obtenues par des modifications de l'environnement système sont également possibles. Elles ne sont pas décrites ici, car elles sont liées aux spécificités de chaque site et ne relèvent pas de principes généraux pouvant s'appliquer à tous les sites.

De plus, il est conseillé de modifier le moins possible les options par défaut choisies afin que la reprise de ces modifications lors des réimplantations ne constitue pas une lourde charge pour le gestionnaire des bases VA Pac.

3.8. INSTALLATION DES FICHIERS INDEX ET DONNEES

INSTALLATION DES FICHIERS INDEX (AN) ET DONNEES (AR)

AN et AR doivent être autant que possible installés sur des VOLUMES DIFFERENTS.

Pour les bases VA Pac importantes (à partir d'un million de données), il peut également être intéressant de répartir les fichiers AN et AR sur plusieurs volumes.

Le fichier AN étant sollicité dans son ensemble par l'activité VA Pac, il suffit d'adapter le DEFINE du fichier dans le membre DFxxnnAN du PDS des paramètres de la base Pacbase, pour une implantation multi-volumes.

En revanche, 80% environ de l'activité VA Pac sollicite les données rangées en DEBUT du fichier AR, dans le cas du travail standard d'une équipe de développement d'applications. La répartition du fichier AR sur plusieurs volumes n'apporte donc pas en elle-même de possibilité d'optimisation des accès.

Une procédure standard, STOP ("Storage Optimization"), permet d'améliorer cette répartition multi-volumes, par l'éclatement des données les plus fréquemment activées dans l'ENSEMBLE du fichier.

Cette procédure, combinée avec le DEFINE sur plusieurs volumes du fichier AR (membre DFxxnnAR du PDS SY), doit donc être utilisée avant restauration de la base.

(Voir le manuel "LES PROCEDURES BATCH : GUIDE DE L'ADMINISTRATEUR" pour la description de la procédure STOP et le paramétrage de l'éclatement des données.)

3.9. STRUCTURE DE VISUALAGE PACBASE SOUS CICS

STRUCTURE DE VA Pac SOUS CICS

De façon générale, chaque type de ligne VA Pac est traité par un programme défini sous CICS. Il faut distinguer deux types de programmes :

- les programmes pouvant effectuer une mise à jour de la base,
- les programmes qui ne font que lire la base.

A une transaction VA Pac peuvent correspondre plusieurs programmes. Le premier chargé, appelé 'moniteur', passe le contrôle par XCTL au programme apte à traiter le type de ligne affiché à l'écran. Celui-ci exécute alors les instructions suivantes :

. Lecture de l'écran (RECEIVE) :

. Si détection de mise à jour :

- exécution d'un ENQUEUE sur une chaîne de caractères propre à la base mise à jour ;
- READ UPDATE du premier enregistrement du fichier AR ;
- REWRITE de cet enregistrement ;
- boucle sur les mises à jour demandées pouvant comporter par ligne mise à jour :

- . plusieurs positionnements ou lectures séquentielles sur le fichier AN (STARTBR, RESETBR ou READNEXT),
- . plusieurs lectures simples du fichier AR (READ),
- . un ou deux ordres d'écriture sur le fichier AR

(WRITE ou READ UPDATE, REWRITE),

- . plusieurs ordres d'écriture sur le fichier AN

(WRITE ou READ UPDATE, REWRITE)

- . un ordre d'écriture sur le fichier AJ (WRITE);

- READ UPDATE du premier enregistrement du fichier AR ;
- REWRITE de cet enregistrement ;
- exécution d'un DEQUEUE correspondant à l'ENQUEUE.

. Si un choix est exprimé : LINK au programme de décodage de la zone choix ;

. Si le choix est valide : XCTL sur le programme correspondant ;

. Phase d'affichage se décomposant en :

- plusieurs positionnements ou lectures séquentielles sur le fichier AN (STARTBR, RESETBR ou READNEXT) ;
- plusieurs lectures simples du fichier AR (READ) ;

. Envoi de l'écran (SEND) ;

. Fin de la transaction (RETURN TRANSID) ;

REMARQUES

Le mécanisme des ENQUEUE/DEQUEUE est fait de telle sorte que les mises à jour sont sérialisées ;

La structure des programmes est telle que, à un instant donné, un seul BROWSING ou un seul READ UPDATE est actif sur le fichier AN; un seul string est consommé par une transaction pendant un laps de temps supérieur à une entrée-sortie.

3.10. RESSOURCES NECESSAIRES SOUS CICS

RESSOURCES NECESSAIRES SOUS CICS

Pour améliorer les performances de VA Pac sous CICS, deux solutions sont à considérer :

- diminuer les temps de chargement des programmes,
- rendre plus efficace la gestion des buffers VSAM.

1. La gestion des programmes

Le premier point est relativement facile à satisfaire : il suffit d'observer la fréquence d'appel des différents programmes et de les déclarer résidents dans la partition CICS (RESIDENT(YES) en CSD).

A titre d'exemple, les programmes suivants sont les plus fréquemment utilisés :

xxR000	moniteur
xxR100	décodage de la zone choix
xxR200	contrôle des formats
xxQ300 xxN300	mise à jour lignes -CE d'un segment
xxQT00 xxNT00	mise à jour lignes -D d'une tâche
xxQP00 xxNP00	mise à jour lignes -P
xxQ700 xxN700	mise à jour lignes -W
xxQI00 xxNI00	mise à jour lignes -CE d'un dialogue
xxQI01 xxNI01	-
xxQI02 xxNI02	-
xxQH20 xxNH20	mise à jour lignes -CS d'un dialogue

2. La gestion des buffers

Trois paramètres associés aux fichiers AN et AR sont à renseigner en CSD :
STRINGS, DATABUFFERS et INDEXBUFFERS.

- STRINGS :

Il spécifie le nombre de STRINGS ou requêtes pouvant être exécutées en parallèle par VSAM sur un même fichier : normalement, un STRING est consommé le temps de l'entrée-sortie sur disque.

Cependant, certains ordres échappent à cette règle :
READ UPDATE et STARTBR ou RESETBR réservent un STRING jusqu'aux ordres REWRITE ou UNLOCK et ENDBR respectivement.

- DATABUFFERS :

Il spécifie le nombre de buffers DATA devant être alloués par VSAM à l'ouverture du fichier. Un minimum de deux buffers est nécessaire : un pour la lecture du "control interval DATA", un pour le 'split' de "control interval".
Par défaut : DATABUFFERS = STRINGS + 1

- INDEXBUFFERS :

Il spécifie le nombre de buffers INDEX devant être alloués par VSAM à l'ouverture du fichier. Le nombre minimum de buffers est de 1 par requête.
Il est toutefois préférable d'augmenter ce nombre car, pour un fichier ayant trois niveaux d'INDEX, trois EXCP sont nécessaires pour pointer le CI DATA recherché pour chaque requête en accès direct.

Il est conseillé d'allouer :

- au minimum : un buffer par niveau d'INDEX
+ un buffer par STRING (valeur de STRINGS)
- au maximum : un buffer par CI de l'INDEX SET
+ un buffer par STRING.

Conclusion

Il est important de prévoir un nombre de buffers suffisant en fonction du nombre de STRINGS.

Il peut être préjudiciable de réserver un nombre trop important de buffers car cela peut entraîner une pagination excessive, donc des EXCP.

Nous proposons ci-dessous un tableau des valeurs optimum des différents paramètres en fonction du nombre de STRINGS :

```

-----
! Nombre de !      Fichier AN      ! Fichier AR  !
! STRINGS  !-----!
!          !STRINGS! INDEX ! DATA !STRINGS! DATA !
!          !      !BUFFERS!BUFFERS!      !BUFFERS!
!-----!
!      1      !      1      !      4      ! > 4      !      1      !      2      !
!-----!
!      5      !      5      !      8      !      7      !      5      !      6      !
!-----!
!     10      !     10      !     13      !     11      !     10      !     11      !
!-----!
!     20      !     20      !     23      !     21      !     20      !     21      !
!-----!
!     30      !     30      !     33      !     31      !     30      !     31      !
!-----!
!     40      !     40      !     43      !     41      !     40      !     41      !
-----

```

Le problème restant à résoudre est celui de la détermination du nombre de requêtes concurrentes à un instant donné.

3.11. DETERMINATION DU NOMBRE DES REQUETES SOUS CICS

DETERMINATION DU NOMBRE DE REQUETES SOUS CICS

Multiplier le nombre de STRINGS nécessaires à une transaction par le nombre d'utilisateurs maximum aboutirait à allouer un trop grand nombre de buffers et demanderait un espace mémoire trop important.

Nous avons préféré adopter une démarche probabiliste en partant de quelques hypothèses simples mais représentatives des mécanismes de gestion des ressources par CICS.

La méthode est la suivante :

On suppose que la loi d'arrivée des demandes d'exécution des transactions est poissonnienne et que la durée d'une transaction est fixe.

A partir de cette durée et de la fréquence moyenne des demandes, on détermine le nombre N, tel que la probabilité d'observer plus de N transactions à un instant donné soit inférieure à 1 / 100 000.

Il ne reste plus alors pour un site particulier qu'à déterminer la durée moyenne d'une transaction et leur fréquence moyenne pour, à l'aide du tableau suivant trouver ce nombre N.

Ce nombre est égal au nombre de STRINGS indiqué dans le tableau précédent qui permet de déterminer les paramètres STRINGS, DATABUFFERS et/ou INDEXBUFFERS pour les fichiers AN et AR.

Les temps sont exprimés en secondes, les fréquences moyennes variant de 0,1 à 2, ce qui correspond à des périodes variant de 10 s. à 0,5 s.

Voir le tableau page suivante.

```

-----
!Freq.!Temps                                     !
!      !                                         !
!      !0,1 0,5 0,9 1,3 1,7 2,1 2,5 2,9 3,3 3,7 4,1 4,5 4,9 !
!-----!
! 0,1 ! 2 3 4 4 5 5 5 5 6 6 6 6 6 !
! 0,2 ! 3 4 5 5 6 6 7 7 7 8 8 8 8 !
! 0,3 ! 3 4 5 6 7 7 8 8 9 9 9 10 10 !
! 0,4 ! 3 5 6 7 7 8 9 9 10 10 11 11 12 !
! 0,5 ! 3 5 6 7 8 9 9 10 11 11 12 12 13 !
! 0,6 ! 3 5 7 8 9 9 10 11 12 12 13 14 14 !
! 0,7 ! 3 6 7 8 9 10 11 12 13 13 14 15 15 !
! 0,8 ! 4 6 7 9 10 11 12 13 13 14 15 16 16 !
! 0,9 ! 4 6 8 9 10 11 12 13 14 15 16 17 17 !
! 1,0 ! 4 7 8 10 11 12 13 14 15 16 17 18 19 !
! 1,1 ! 4 7 9 10 11 13 14 15 16 17 18 19 20 !
! 1,2 ! 4 7 9 10 12 13 14 15 16 18 19 20 21 !
! 1,3 ! 4 7 9 11 12 14 15 16 17 18 19 20 22 !
! 1,4 ! 4 7 9 11 13 14 15 17 18 19 20 21 22 !
! 1,5 ! 4 8 10 12 13 15 16 17 19 20 21 22 23 !
! 1,6 ! 4 8 10 12 14 15 17 18 19 21 22 23 24 !
! 1,7 ! 5 8 10 12 14 16 17 19 20 21 23 24 25 !
! 1,8 ! 5 8 11 13 14 16 18 19 21 22 23 25 26 !
! 1,9 ! 5 8 11 13 15 17 18 20 21 23 24 26 27 !
! 2,0 ! 5 9 11 13 15 17 19 20 22 24 25 26 28 !
-----
    
```

3.12. ANOMALIES D'EXECUTION

ANOMALIES D'EXECUTION

Il arrive que des anomalies se produisent lors de l'exécution d'un programme batch. En particulier, les erreurs d'entrée-sortie sur les fichiers du système ou de la base provoquent une fin anormale par un ABEND USER, accompagné d'un message émis sur le fichier SYSLST.

En présence d'un ABEND, l'utilisateur doit avant tout rechercher la présence de ce message :

```
PROGR : pppppp INPUT-OUTPUT ERROR : FILE ff OP : oo  
STATUS : nn  
END OF RUN DUE TO PROVOKED ABEND
```

Dans la plupart des cas, l'examen du "status" et le type d'opération effectuée permettent de trouver la cause de la fin anormale.

Quelques valeurs courantes de 'STATUS' et de 'OP' :

```
+-----+-----+ +-----+-----+  
! oo ! opération ! ! nn ! STATUS !  
+-----+-----+ +-----+-----+  
! ! ! ! 21 ! Erreur de séquence !  
! W ! WRITE ! ! 22 ! Clé en double !  
! RW ! REWRITE ! ! 23 ! Enregistrement non trouvé !  
! RU ! READ UP ! ! 24 ! Fichier trop petit (KSDS-RRDS) !  
! OP ! OPEN ! ! 30 ! Erreur système !  
! CL ! CLOSE ! ! 30 ! Erreur système !  
! D ! DELETE ! ! 34 ! Fichier trop petit (séquentiel) !  
! R ! READ ! ! 92 ! Erreur logique (par exemple, !  
! P ! START ! ! ! ouverture fichier déjà ouvert) !  
! RN ! READ NEXT ! ! 93 ! Fichier resté ouvert sous CICS !  
! ! ! ! 95 ! Fichier non défini/mal défini !  
+-----+-----+ +-----+-----+
```

Si le message est absent et que le type de l'ABEND met en cause directement les programmes du système VA PAC, il est nécessaire de contacter l'équipe technique VA PAC et de conserver tous les listings qui lui seront nécessaires pour analyser le problème :

.Le compte-rendu d'exécution avec // OPTION LOG,PARTDUMP
.La liste des programmes et fichiers installés (job 9INS)
.Etc.

REPRISE APRES ARRET ANORMAL D'UNE PROCEDURE

En cas d'arrêt d'une procédure pour une raison telle que :

- fichier en sortie trop petit,
- partition trop petite pour l'ouverture d'un fichier,
- fichier ouvert dans une autre partition,
- d'une manière générale, toutes les erreurs ne tenant pas aux programmes de traitement ni aux données traitées,

il est possible de relancer la procédure à partir du step où l'arrêt s'est produit après correction de l'erreur.

Cette possibilité, dont la mise en oeuvre exige une certaine vigilance, peut permettre l'économie de temps de traitement.

La ligne de JCL '/* GOTO JCLSTxx' doit être activée, sans la déplacer, avec le label du step où la fin anormale s'est produite.

Certaines précautions sont à observer :

- Si les noms physiques des fichiers temporaires sont préfixés par le caractère '%', veiller à soumettre le job de reprise sous la même partition que le job initial, ou à substituer dans le job de reprise les noms des fichiers temporaires, en supprimant le préfixe '%' et en ajoutant le suffixe '.BG', '.F2', '.F3', etc. selon que le job initial aura été soumis sous BG, F2, F3, etc.
- Si la fin anormale est due à un fichier trop petit en sortie, il faudra veiller en le redéfinissant à ne pas toucher aux fichiers temporaires en attente.

Vous pouvez déterminer ceux-ci en vous référant attentivement à la description de la procédure en cause :

- aux fichiers en entrée de l'étape où la fin anormale s'est produite,
- éventuellement aux fichiers en entrée d'une des étapes suivantes qui ne seraient en sortie d'aucune des étapes de la partie de procédure à resoumettre, compte tenu de la chronologie d'exécution.

4. INSTALLATION

4.1. PREAMBULE

NOUVEAUTES ET MODIFICATIONS

Afin de faciliter la mise en oeuvre de cette version de VA PAC pour les utilisateurs des versions 7.3, 8.0 ou 8.0.1, vous trouverez ci-dessous la liste des principales nouveautés et modifications affectant cette version de VA PAC .

Elles sont classées selon 3 domaines :

- . Habillage : organisation des JCL.
- . Environnement : fichiers modifiés, ajoutés...
- . Contenu : nouvelles procédures...

HABILLAGE

Les fichiers temporaires VSAM utilisent désormais l'option 'NOALLOCATION', ce qui permet d'alléger les JCL's en reportant tous les DELETE/DEFINE en tête de procédure.

Le renvoi d'un code-retour au superviseur par les programmes COBOL permet la gestion de celui-ci dans les JCL's et la fusion de procédures auparavant distinctes (PARM et PAR2, MLIB et MLIS...).

L'utilisation des procédures POWER (membres SLI) permet d'une part, la non-redondance des cartes de commande IDCAMS pour les DEFINE CLUSTER des fichiers de la base qui sont désormais centralisés dans des membres SLI, d'autre part le catalogage de procédures lourdes comme GPRT. Une procédure dite 'début de job standard' est également mise en oeuvre dans tous les JCL's afin d'éviter la dispersion des standards non prévus spécifiques à chaque site.

Les procédures VSE sont utilisées pour la définition des espaces de travail (essentiellement dans les tris, mais aussi dans les rares cas où les fichiers de travail VSAM sont inadaptés). La mise en oeuvre de VSAM SPACE MANAGEMENT pour les fichiers de travail des tris ne pose plus de problèmes.

Pour faciliter la gestion des fichiers de sauvegarde, et également permettre le choix du nombre de 'générations' conservées pour chacun d'eux, les cartes de commandes IDCAMS effectuant la rotation des générations sont centralisées dans des membres SLI.

ENVIRONNEMENT

Apparition de deux nouveaux fichiers utilisateurs : le fichier PA (fichier de travail de PAF TP) et le fichier GS (fichier des schémas d'édition de PAF-PDM).

Apparition d'une 'enveloppe' de fichier de travail par partition batch (fichier de travail de PAF batch).

Apparition de trois fichiers systèmes : SS, SF et SP (squelettes de génération).

Le JCL de lancement de POWER doit être prêt pour la prise en compte des procédures SLI.

La variante VSE (version système antérieure à VSE/SP 2) n'est plus portée.

Le paramétrage initial (MM1JCL) offre la possibilité de répartir les composants 'index' et 'data', les fichiers 'utilisateur' et 'système', les fichiers de sauvegarde et d'exploitation, les fichiers permanents et temporaires, sur autant de volumes disques différents.

CONTENU

Nouvelles procédures : CRYP, LVBL, UPAE, XPAF, XPDM, XUSR, YSMC

Nouveau générateur batch.

Procédures supprimées : MLIS, PAR2

4.2. PRESENTATION GENERALE

PRESENTATION GENERALE

La procédure d'installation est composée de trois grandes phases :

- . Préparation de l'implantation,
- . Implantation,
- . Tests conversationnels et batch.

Elle utilise une cartouche (ou bande) d'implantation et son déroulement est décrit dans le présent chapitre.

Avant de procéder à l'implantation, l'utilisateur doit avoir pris connaissance des caractéristiques techniques du système VA Pac décrites dans le présent manuel, afin de prévoir l'environnement nécessaire au bon déroulement de l'implantation (place sur disques, options de CICS, codification, etc.).

L'installation peut alors être faite et son déroulement est le suivant :

PREPARATION

Récupération du JCL initial présent sur la cartouche et exécution de ce JCL :

- . Sauvegarde de la cartouche d'implantation,
- . Copie d'un module de traitement du JCL complet,
- . Récupération du JCL complet d'implantation et d'exploitation de VisualAge Pacbase.

IMPLANTATION

(Voir sous-chapitre "Déroulement de l'Implantation").

TESTS

- . Tests conversationnels,
- . Tests des procédures batch.

4.3. BANDE D'IMPLANTATION

BANDE D'IMPLANTATION

La bande d'implantation (6250 BPI, sans label) contient les fichiers suivants :

!Rang!	Format!	Lrecl!	Blksize!	Contenu	!
! 1 !	Fixe !	80 !	3440 !	JCL initial de préparation	!
! 2 !	Fixe !	80 !	80 !	Load-module de l'utilitaire d'installation MM1JCL	!
! 3 !	Fixe !	80 !	3440 !	JCL complet d'installation et d'exploitation	!
! 4 !	Fixe !	3204 !	32040 !	Fichier squelette de génération Langage Batch	!
! 5 !	Fixe !	4605 !	32235 !	Fichier squelette de génération Dialogue et DBD	!
! 6 !	Fixe !	4605 !	32235 !	Fichier squelette de génération Cobol	!
! 7 !	Fixe !	80 !	32720 !	Libellés d'erreur PACBASE	!
! 8 !	Varia!	155 !	32760 !	Sauvegarde de la base de tests	!
! 9 !	Fixe !	4605 !	23025 !	Fichier squelette de génération Extension PAF-PDM	!
! 10 !	Fixe !	119 !	11900 !	Fichier squelette de génération Extension PAF-PDM	!
! 11 !	Fixe !	4605 !	32235 !	Fichier squelette C/S	!
! 12 !	Fixe !	3204 !	32040 !	Squelette batch version angl.	!
! 13 !	Fixe !	4605 !	32235 !	Squelette TP version angl.	!
! 14 !	Fixe !	4605 !	32235 !	Squelette cobol version angl.	!
! 15 !	Varia!	155 !	32760 !	Base de tests version angl.	!

!Rang!	Format!	Lrecl!	Blksize!	Contenu	!
! 16 !	Fixe !	80 !	32720 !	AE0 initial, version bilingue	!
! 17 !	Fixe !	4605 !	32235 !	Squelette C/S version angl.	!
! 18 !	Fixe !	80 !	32720 !	Source programme pour SQL	!
! 19 !	Fixe !	117 !	7020 !	Dictionnaire Visualage	!
! 20 !	Undef!	!	----- !	Fichier vide (Backup header)	!
! 21 !	Undef!	!	17000 !	Backup file-id	!
! 22 !	Undef!	!	17000 !	Backup file	!
! 23 !	Undef!	!	----- !	Fichier vide (Backup trailer)	!
! 24 !	Fixe !	80 !	80 !	End-of-backup record	!

4.4. DECHARGEMENT DU JCL INITIAL

GENERATION DU JCL INITIAL

Le JCL initial se trouve dans le premier fichier de la bande d'implantation et doit être déchargé sur l'éditeur du site. (LRECL 80 BLKSIZE 80)

I N S T A L L A T I O N
DECHARGEMENT DU JCL INITIAL

PAGE

62

4
4

```
* $$ JOB JNM=$JOB,CLASS=$CLASS,DISP=$DISP,USER=' $USER '  
* $$ LST CLASS=$PCLO,DISP=$PDIO  
* $$ PUN CLASS=$PCLP,DISP=$PDIP,DEST=$DEST  
// JOB $JOB          DECHARGEMENT DU JCL INITIAL PACBASE  
// ASSGN SYS004,TAPE  
// PAUSE MONTER LA BANDE D'IMPLANTATION PACBASE SUR SYS004  
// UPSI 1  
// EXEC DITTO  
$$DITTO REW OUTPUT=SYS004  
$$DITTO TC INPUT=SYS004,NFILES=1  
$$DITTO RUN OUTPUT=SYS004  
$$DITTO EOJ  
/*  
/&  
* $$ EOJ
```

4.5. EXECUTION DU JCL INITIAL

JCL INITIAL

Le JCL initial a pour but de générer le JCL d'installation et d'exploitation du système qui est livré sous la forme d'un squelette paramétré permettant l'adaptation aux besoins de l'utilisateur. Ce squelette est manipulé par un utilitaire ('MM1JCL') générant un JCL adapté après choix des paramètres appropriés.

Le déchargement de cet utilitaire et son utilisation sont fournis dans le JCL initial qui contient trois JOBS :

JOB0 Copie de la bande d'implantation livrée par IBM sur une bande de l'utilisateur : c'est cette dernière, constituant la sauvegarde du système PACBASE pour l'utilisateur, qui doit être utilisée pour l'implantation.

JOB1 Déchargement de l'utilitaire MM1JCL et des programmes système (par fonction RESTORE de LIBR).

JOB2 Exécution de MM1JCL pour installer le JCL d'implantation et d'exploitation de PACBASE. Cette exécution doit être conservée ; elle peut être réutilisée pour des réimplantations.

```
/* -----  
/* J C L       I N I T I A L       D ' I M P L A N T A T I O N  
/* -----  
/*  
/* CE JCL CONTIENT TROIS JOBS :  
/*   - COPIE DE LA BANDE DE REFERENCE SUR UNE BANDE  
/*     D'IMPLANTATION,  
/*   - RECEPTION DE L'UTILITAIRE D'IMPLANTATION  
/*     DANS VOTRE LIBRAIRIE,  
/*   - RECEPTION DES JCL D'IMPLANTATION ET D'EXPLOITATION  
/*     PAR VOTRE EDITEUR DE TEXTE.  
/*  
/* LES ELEMENTS VARIABLES QUE CES TROIS JOBS UTILISENT SONT :  
/*   - UN CATALOGUE VSAM POUR LES FICHIERS DE TRAVAIL,  
/*   - UNE LIBRAIRIE POUR L'UTILITAIRE,  
/*   - LE SPOOL DE POWER POUR LE JCL PARAMETRE EN SORTIE,  
/*   - UN CODE USER POUR POWER,  
/*   - DEUX NOMS DE BANDES,  
/*   - DEUX ADRESSES DE DEROULEURS.  
/*  
/* ----- SUBSTITUTIONS A EFFECTUER -----  
/*  
/* %USER = IDENTIFIANT JOB SUR 4 CAR  
/* %ACCT = ACCOUNTING POUR POWER (USER='%ACCT')  
/*  
/* %CATN = IDENTIFICATION DU CATALOGUE VSAM  
/* %CATV = VOLUME SERIAL NUMBER ESPACE VSAM  
/*  
/* %LIBN = IDENTIFICATION D'UNE LIBRAIRIE POUR UTILITAIRE  
/* %LIB  = LIB POUR UTILITAIRE  
/* %SLIB = SUBLIB POUR UTILITAIRE  
/*  
/* %PLST = PARAMETRES * $$ LST  
/* %PPCH = PARAMETRES * $$ PUN  POUR RECUPERATION SOUS EDITEUR  
/*  
/* %INST = VOLUME SERIAL NUMBER DE LA BANDE DE REFERENCE  
/* %BCKP = VOLUME SERIAL NUMBER DE LA SAUVEGARDE  
/*  
/* %TAP1 = ADRESSE DU PREMIER DEROULEUR  
/* %TAP2 = ADRESSE DU DEUXIEME DEROULEUR  
  
/* *****  
/*          COPIE DE LA BANDE D'INSTALLATION  
/* *****  
* $$ JOB JNM=%USERJOB0,CLASS=0,DISP=D,USER='%ACCT'  
* $$ LST %PLST  
// JOB JOB0          COPIE DE LA BANDE D'INSTALLATION  
// PAUSE MOUNT      : %INST,%TAP1,READ    %BCKP,%TAP2,WRITE  
// UPSI 1  
// EXEC DITTO  
$$DITTO REW OUTPUT=%TAP1  
$$DITTO REW OUTPUT=%TAP2  
$$DITTO TT INPUT=%TAP1,OUTPUT=%TAP2,NFILES=24  
$$DITTO RUN OUTPUT=%TAP1  
$$DITTO EOJ  
/*  
/&  
* $$ EOJ  
  
* $$ JOB JNM=%USERJOB1,CLASS=0,DISP=D,USER='%ACCT'  
* $$ LST %PLST  
// JOB JOB1          COPIE DE L'UTILITAIRE  
/* *****  
/* *      DECHARGEMENT DE L'UTILITAIRE D'INSTALLATION MMLJCL  
/* *      DANS LA BIBLIOTHEQUE DE L'UTILISATEUR PREDEFINIE.  
/* *****  
// PAUSE MONTER LA BANDE D'IMPLANTATION %BCKP SUR %TAP2
```

INSTALLATION
EXECUTION DU JCL INITIAL

4
5

```

MTC REW,%TAP2
MTC FSF,%TAP2,1
// ASSGN SYSIPT,%TAP2
// DLBL %LIB,'%LIBN',,VSAM
// LIBDEF PHASE,CATALOG=%LIB.%SLIB
// OPTION CATAL
INCLUDE
// EXEC LNKEDT
/*
/&
* $$ EOJ

* $$ JOB JNM=%USERJOB2,CLASS=0,DISP=D,USER='%ACCT'
* $$ LST %PLST
* $$ PUN %PPCH
// JOB JOB2          CREATION DU JCL D'INSTALLATION
/* *****
* --- CREATION DU JCL D'INSTALLATION PAR L'UTILITAIRE MM1JCL
/* *****
/*
/* .MODIFIER LA LISTE DES COMMANDES FOURNIES EN PRECISANT LES
/* VARIANTES D'INSTALLATION (S'IL Y EN A), EN DEMANDANT SI
/* NECESSAIRE LA SELECTION DE PORTIONS DE JCL D'INSTALLATION
/* (MODULES DE JCL), EN DONNANT DES VALEURS APPROPRIEES AUX
/* PARAMETRES D'INSTALLATION, ET EN PRECISANT EVENTUELLEMENT
/* DES LIGNES A AJOUTER EN TETE OU EN FIN DES MODULES.
/*
/* *****
* --- STEP 1 --- IDCAMS DEFINE
// DLBL IJSYSUC,'%CATN',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          (PTR.IMPL.PAC7GC) CL
DEF CL      (NAME (PTR.IMPL.PAC7GC) -
            NAL RUS NIXD CISZ(4096) -
            RECFM(FB(104)) -
            VOL (%CATV) RECSZ (3432 3432) TRK (10 5) )
DEL          (PTR.IMPL.PAC7GI) CL
DEF CL      (NAME (PTR.IMPL.PAC7GI) -
            NAL RUS NIXD CISZ(4096) -
            RECFM(FB(104)) -
            VOL (%CATV) RECSZ (3432 3432) TRK (50 10) )
DEL          (PTR.IMPL.PAC7LA) CL
DEF CL      (NAME (PTR.IMPL.PAC7LA) -
            NAL RUS NIXD CISZ(4096) -
            RECFM(FB(137)) -
            VOL (%CATV) RECSZ (1370 1370) TRK (5 1) )
DEL          (PTR.IMPL.PAC7LI) CL
DEF CL      (NAME (PTR.IMPL.PAC7LI) -
            NAL RUS NIXD CISZ(4096) -
            RECFM(FB(137)) -
            VOL (%CATV) RECSZ (1370 1370) TRK (5 1) )
/*
* --- STEP 2 --- MM1JCL
// ASSGN SYS004,%TAP2
// PAUSE MONTER LA BANDE D'INSTALLATION %BCKP EN %TAP2
// MTC REW,SYS004
// MTC FSF,SYS004,2
// ASSGN SYS005,SYSLST
// ASSGN SYS010,SYSIPT
// ASSGN SYS011,SYSPCH
// DLBL IJSYSUC,'%CATN',,VSAM
// DLBL %LIB,'%LIBN',,VSAM
// LIBDEF PHASE,SEARCH=%LIB.%SLIB
// DLBL WK1,'PTR.IMPL.PAC7GC',,VSAM
// DLBL WK2,'PTR.IMPL.PAC7GI',,VSAM
// DLBL LBI,'PTR.IMPL.PAC7LA',,VSAM
// DLBL LBO,'PTR.IMPL.PAC7LI',,VSAM
/*      LE JCL GENERE SORT NECESSAIREMENT EN PUNCH
/*      CE JCL EST A RECUPERER SOUS UN EDITEUR DE TEXTE
// EXEC MM1JCL,SIZE=AUTO

```

```

*
F
===IDENC CLIENT 'TOUS CLIENTS',SITE 'JCL PROTOTYPE'
===SELV MER .STATION METHODE MERISE
===SELV YSM .STATION METHODE YSM
===SELV SSA .STATION METHODE SSADM
===SELV IFW .STATION METHODE IFW
===SELV OMT .STATION METHODE OMT
===SELV PEI .PACBASE ENVIRONNEMENT DE PRODUCTION
===SELV PACTR .PACBASE PAC/TRANSFER
===SELV PUF .PACBASE PUFTP
===SELV SQL .PACBASE SQL
===SELV S2K .PACBASE Pac/Impact
===SELV TAB .PACBASE TABLES
===SELV DSM .PACBASE DSMS
===SELV TEAM .PACBASE PONT TEAMCONNECTION
===SELV VISP .PACBASE PONT VISUALAGE SMALLTALK
===SELV ROLD .REPRISE ANCIENS PACBASE
===PRM PCJ=0 .CLASSE D'EXECUTION (CARTE JOB)
===PRM PCL=X .CLASSE DE SORTIE (CARTE LST)
===PRM PCP=X .CLASSE DE SORTIE (CARTE PUN)
===PRM PDJ=D .DISPOSITION DES JOBS (CARTE JOB)
===PRM PDL=H .DISPOSITION DE SORTIE (CARTE LST)
===PRM PDP=H .DISPOSITION DE SORTIE (CARTE PUN)
===PRM LST2=60C .IMPRIMANTE 2 (POWER PARAMETRE LST=---)
===PRM LST3=60D .IMPRIMANTE 3 (POWER PARAMETRE LST=---)
===PRM PRFJ=%USER .PREFIXE NOMS DE JOBS (4 CARACTERES)
===PRM PUJ='%ACCT' .PARAMETRE USER (CARTE JOB)
===PRM PWJ='' .PARMS SUPPLEMENTAIRES CARTE JOB POWER
===PRM PWL=',RBS=250' .PARMS SUPPLEMENTAIRES CARTE LST POWER
===PRM PWP='' .PARMS SUPPLEMENTAIRES CARTE PUN POWER
===PRM GROUP='PACB' .CICS/RDO : GROUPE RESSOURCES PACBASE
===PRM LIST='LCGI' .CICS/RDO : LISTE RESSOURCES PACBASE
===PRM NLSR=1 .NUMERO DE POOL LSR
===PRM PFI=25 .NUMERO BASE UTILISATEUR (2 CARACTERES)
===PRM PRO=PB .RADICAL DU SYSTEME (2 CARACTERES)
===PRM PXU='PAC250.USR' .INDEX DES FICHIERS PACBASE UTILISATEUR
===PRM PXY='PAC250.SYS' .INDEX DES FICHIERS PACBASE SYSTEME
===PRM TAPEI=%BCKP .NOM DE LA BANDE D'INSTALLATION
===PRM TAP1=%TAP1 .ADRESSE D'UN DEROULEUR
===PRM TAP2=%TAP2 .ADRESSE D'UN 2EME DEROULEUR
===PRM VOLVWK=%CATV .VOLUME DES FICHIERS TEMPORAIRES/DYNAM.
===PRM VWORK=%CATV .VOLUME DES FICHIERS DE TRAVAIL
===PRM XW=%WPAC .INDEX FICHIERS DE TRAVAIL
===PRM WKBG1='12875,400' .WORK AREA 1/BG (400 TRACKS 3380)
===PRM WKBG2='12475,400' .WORK AREA 2/BG
===PRM WKBG3='12075,400' .WORK AREA 3/BG
===PRM WKF21='0,0' .WORK AREA 1/F2 - FACULTATIF
===PRM WKF22='0,0' .WORK AREA 2/F2 "
===PRM WKF23='0,0' .WORK AREA 3/F2 "
===PRM WKF31='0,0' .WORK AREA 1/F3 "
===PRM WKF32='0,0' .WORK AREA 2/F3 "
===PRM WKF33='0,0' .WORK AREA 3/F3 "
===PRM WKF41='11675,400' .WORK AREA 1/F4 "
===PRM WKF42='11275,400' .WORK AREA 2/F4 "
===PRM WKF43='10875,400' .WORK AREA 3/F4 "
===PRM WKF51='10475,400' .WORK AREA 1/F5 "
===PRM WKF52='10075,400' .WORK AREA 2/F5 "
===PRM WKF53='9675,400' .WORK AREA 3/F5 "
===PRM WKF61='0,0' .WORK AREA 1/F6 "
===PRM WKF62='0,0' .WORK AREA 2/F6 "
===PRM WKF63='0,0' .WORK AREA 3/F6 "
===PRM WKF71='0,0' .WORK AREA 1/F7 "
===PRM WKF72='0,0' .WORK AREA 2/F7 "
===PRM WKF73='0,0' .WORK AREA 3/F7 "
===PRM WKF81='0,0' .WORK AREA 1/F8 "
===PRM WKF82='0,0' .WORK AREA 2/F8 "
===PRM WKF83='0,0' .WORK AREA 3/F8 "
===PRM WKF91='0,0' .WORK AREA 1/F9 "
===PRM WKF92='0,0' .WORK AREA 2/F9 "
===PRM WKF93='0,0' .WORK AREA 3/F9 "
===PRM MODB='%LIBN' .NOM PHYSIQUE CIL MODULES BATCH
===PRM MODO='%LIBN' .NOM PHYSIQUE CIL MODULES TP
===PRM SLIL='%LIBN' .NOM PHYSIQUE LIB MEMBRES SLI

```

```
===PRM PRLB='%LIBN' .NOM PHYSIQUE LIB PROCEDURES VSE
===PRM LMODB='%LIB' .LIB / MODULES BATCH
===PRM LMODO='%LIB' .LIB / MODULES TP
===PRM LSLI='%LIB' .LIB / SLI MEMBER
===PRM LPRLB='%LIB' .LIB / VSE PROCEDURES
===PRM SMOB='%SLIB' .SUBLIB / MODULES BATCH
===PRM SMODO='%SLIB' .SUBLIB / MODULES TP
===PRM SLIS='%SLIB' .SUBLIB / SLI MEMBER
===PRM SLIT='PAC' .TYPE / SLI MEMBER
===PRM SPRLB='%SLIB' .SUBLIB / VSE PROCEDURES
===PRM SORTWK='$$SORTWK' .NOM PROCEDURE DEFINITION SORTWORK
===PRM VCAP='%CATN' .CATALOGUE VSAM FICHIERS PACBASE
===PRM OPT='LOG,NODUMP' .PARAMETRES CARTE // OPTION
===PRM RAJP=3000 .ALLOCATION PRIMAIRE DE AJ EN RECORDS
===PRM RAJS=3000 .ALLOCATION SECONDAIRE AJ EN RECORDS
===PRM RPCP=20000 .ALLOCATION PRIMAIRE DE PC EN RECORDS
===PRM RPCS=20000 .ALLOCATION SECONDAIRE PC EN RECORDS
===PRM VOLPSV='%CATV' .VOLUME FICHIERS BACKUP/JOURNAL
===PRM VOLPUD='%CATV' .VOLUME COMPOSANTS DATA FICHIERS USER
===PRM VOLPUX='%CATV' .VOLUME COMPOSANTS INDEX FICHIERS USER
===PRM VOLPYD='%CATV' .VOLUME COMPOSANTS DATA FICHIERS SYST
===PRM VOLPYX='%CATV' .VOLUME COMPOSANTS INDEX FICHIERS SYST
===PRM TRO=TF .RADICAL SYSTEME PACTABLE (2 CAR)
===PRM TXU='TAB250' .INDEX FICHIERS USER TABLES
===PRM VOLTUX='%CATV' .VOLUME COMPOSANT INDEX TABLES USER
===PRM VOLTUD='%CATV' .VOLUME COMPOSANT DATA TABLES USER
===PRM VCAT='%CATN' .CATALOGUE VSAM PACTABLE
===PRM VCAD='%CATN' .CATALOGUE VSAM DSMS
===PRM DSMS=<> .FILE NAME FICHER DSMS (DC)
===PRM MODELF='PAC250.MODEL' .FICHER SAM-ESDS MODELE FIXE
===PRM MODELV='PAC250.MODEL' .FICHER SAM-ESDS MODELE VARIABLE
===BEGMOD
./ ADD NAME=$PRO$PFI$ZMODUL
//%USER$ZMODUL JOB %ACCT,'VSE',CLASS=X,MSGCLASS=X,NOTIFY=%USER
/*XMIT A07POWER DLM=%%
===ENDMOD
%%
/*
* --- STEP 3 --- IDCAMS DELETE
// DLBL IJSYSUC,'%CATN',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL (PTR.IMPL.PAC7GC) CL
DEL (PTR.IMPL.PAC7GI) CL
DEL (PTR.IMPL.PAC7LA) CL
DEL (PTR.IMPL.PAC7LI) CL
/*
/&
* $$ EOJ
```

4.6. INSTALLATION DU JCL COMPLET

INSTALLATION DU JCL COMPLET

Le module MM1JCL lit le fichier de JCL squelette et crée en sortie un JCL adapté à l'aide de commandes permettant de :

- . sélectionner des portions du JCL squelette, nommées modules de JCL,
- . paramétrer le squelette de manière à obtenir un JCL avec un minimum de modifications à effectuer pour le rendre opérationnel,
- . sélectionner des variantes d'installation pour générer le JCL propre à certains traitements spécifiques, en fonction du site ou des conditions d'installation,
- . ajouter des lignes avant et/ou après chaque module de JCL pour séparer les modules.

Cette étape peut être exécutée autant de fois que nécessaire pour produire un JCL adéquat.

MM1JCL fournit une liste par module de JCL créé, avec les paramètres pris en compte et en fonction des variantes demandées.

Une fois le JCL créé, l'utilisateur peut le récupérer sous un éditeur de texte pour entamer l'installation proprement dite.

Il est conseillé de constituer autant de membres dans l'éditeur de texte que de modules de JCL, chaque module constituant une étape logique de l'installation.

Pour automatiser le découpage en modules de JCL, il suffit de choisir avec soin les cartes à ajouter en début et en fin de module. Des exemples en sont donnés dans ce chapitre.

FICHIERS UTILISES PAR MM1JCL

En entrée : squelette du JCL : SYS004 (bande)
commandes de l'utilitaire : SYS010 (reader)
En sortie : JCL généré : SYS011 (punch)
liste du JCL généré : SYS005 (printer)
Travail : WK1 (300000 octets), WK2 (300000 octets),
LBI (dummy) LBO (1 piste)

OPTIONS PAR DEFAUT MM1JCL INSTALLATION

- . Variantes : Toutes.
- . Paramètres : Les valeurs indiquées sont des exemples, elles doivent donc être renseignées.
- . Modules : Aucune sélection, tous les modules sont copiés.

CODIFICATION DES COMMANDES DE MM1JCL

```
===SELL lll          .Sélection de la langue
                      d'installation
                      lll = FRA   français
                          ENG   anglais

===SELV vvvv         .Sélection de variante
                      vvvv = code de variante

===SELM mmmm1 mmmm2 ... .Sélection de modules de JCL
                      mmmm1 = nom de module de JCL
                      mmmm2 = nom de module de JCL ...
                      L'absence de ligne ===SELM
                      sélectionne tous les modules de JCL

===PRM  PPPP=pppp   .Paramètre
                      PPPP = nom du paramètre
                      pppp = valeur du paramètre
```

ATTENTION : Sur les lignes ===PRM ou ===SELV, un commentaire peut être indiqué et il doit obligatoirement être précédé par un point et ne pas dépasser la colonne 72.

```
===BEGMOD           Insertion de lignes en début de module
....1              ) lignes à insérer avant chaque module
....n              )
===ENDMOD           Insertion de lignes en fin de module
....1              ) lignes à insérer après chaque module
....n              )
```

4.7. VARIANTES D'INSTALLATION

TABLEAU DES VARIANTES (===SELV vvv)

! vvv	! Signification	! Type	!
! SQL	! DBD Relationnel SQL ! Installation du source Cobol ! de mise à jour du catalogue SQL	! Module VA Pac	!
! PEI	! Environnement de production ! Installation des fichiers de ! l'environnement de production ! et prise en compte dans VA Pac	! Module VA Pac	!
! TAB	! Tables ! Installation des composants de ! l'interface TABLES-VA Pac	! Module VA Pac	!
! DSM	! DSMS ! Installation des composants de ! l'interface DSMS-VA Pac	! Module VA Pac	!
! PACTR	! Pac/transfer ! Installation des composants de ! l'utilitaire	! Module VA Pac	!
! PUF	! PUFTP ! Installation des programmes de ! PUFTP	! Module VA Pac	!
! S2K	! Pac/Impact ! Installation des composants de ! l'utilitaire	! Module VA Pac	!
! VISP	! Pont visualage smalltalk-pacbase	! Module VA Pac	!
! TEAM	! Pont VAPacbase-TeamConnection	! Module VA Pac	!
! ROLD	! Reprise des bases VA Pac ! 8.0.x, 1.2, 1.6	! Utilisateurs ! 8.0.x, 1.2, 1.6	!

INSTALLATION INTERFACE SQL

Environnement

DB2 n'existe pas sous VSE. Seule l'interface SQL peut donc être disponible.

Présentation du programme

Un programme appelle par LINK un module Cobol contenant les accès SQL :

- xxQS10 génère le SQL de mise à niveau et appelle le module de mise à jour :
 xxRS12.

Installation

Le programme xxQS10 est livré sous forme de module exécutable et installé avec les programmes standard de VA PAC (Job '\$prfj.2PHA' de l'installation).

Le module xxRS12 est livré sous forme de programme source Cobol, pour être préparé, installé et reconnu dans DB2. Pour ce faire, la marche à suivre est la suivante :

1. Le programme source est installé sous un éditeur et l'on vérifie que les ordres correspondent bien aux besoins de l'utilisateur (l'adapter si nécessaire).

Ce source doit être traité comme tout programme SQL classique:

- préparation par préprocesseur donnant

- . le source Cobol,
- . le DBRM (de même nom que le source Cobol).

- compilation et link-edit du source Cobol donnant le programme exécutable (xxRS12).

2. Le plan est construit avec le DBRM (c'est l'opération du BIND).

3. La table RCT de CICS est mise à jour avec le code de la transaction VA PAC et le code du plan précédemment construit.

REMARQUES PARTICULIERES

Les mises à jour sont effectuées à l'aide d'ordres SQL dynamiques.

COMPLEMENT : INSTALLATION APPLICATION INTERACTIVE SQL

Présentation des programmes

Un programme appelle par LINK un module Cobol contenant les ordres SQL. Ce module est le xxQL46, qui génère le SQL de mise à niveau et appelle le module de mise à jour SQL : xxRS12.

Installation

Le programme xxQL46 est livré sous forme de module exécutable et installé avec les programmes standard de VA PAC (Job '\$prfj.2PHA' de l'installation).

Le module xxRS12 est livré sous forme de programme source Cobol, pour être préparé et installé.

4.8. LISTE DES MODULES DE JCL

TABLEAU DES MODULES DE JCL D'INSTALLATION :

Les modules de JCL sont identifiés par un code à 4 positions. Les modules servant à l'installation sont repérés par un préfixe numérique.

La commande ===SELM de l'utilitaire MMIJCL permet de sélectionner les modules à télécharger. Par défaut, tous les modules de JCL sont sélectionnés.

===SELM aaaa bbbb cccc

```

-----
! Code ! Contenu                                     ! Nature !
!-----!-----!-----!
!      !      !      !
! 1CSD ! Source CSD : tables CICS (PCT,PPT)           ! SYS CICS!
! 1PCT ! Source assembleur : transactions (PCT)       ! SYS CICS!
! 1PPT ! Source assembleur : programmes (PPT)        ! SYS CICS!
! 1FCT ! Source assembleur : fichiers (FCT)         ! SYS CICS!
! 1VSE ! Catalogage des procédures VSE                ! JCL VSE !
! 1SLI ! Catalogage des membres SLI                  ! JCL VSE !
! 1PWR ! Complément au JCL de lancement de POWER    ! JCL VSE !
!      !      !      !
! 2FIC ! Définition des fichiers permanents VSAM    ! JCL VSE !
! 2GPR ! Catalogage du JCL de GPRT (membre SLI)     ! JCL VSE !
! 2PAC ! Catalogage du JCL de PACX (membre SLI)     ! JCL VSE !
! 2WGP ! Définition des fichiers de travail GPRT    ! JCL VSE !
! 2WPA ! Définition des fichiers de travail PACX    ! JCL VSE !
! 2PHA ! Installation des modules exécutables       ! JCL VSE !
!      !      !      !
! 3SYS ! Initialisation des fichiers systèmes       ! JCL VSE !
! 3USR ! Initialisation des fichiers utilisateurs   ! JCL VSE !
!      !      !      !
! PJ16 ! Reprise journal archivé                    ! JCL VSE !
! PP16 ! Reprise sauvegarde env. production        ! JCL VSE !
! RPPG ! Reprise commandes édition-génération      ! JCL VSE !
!      !      !      !
-----

```

TABLEAU DES MODULES DE JCL D'INSTALLATION (SUITE) :

```

-----
! Code ! Contenu ! Nature !
!-----!-----!-----!
! ! ! ! !
! 5TPC ! Installation de la base de tests VA PAC ! JCL VSE !
! ! ! ! !
! 6JCI ! Compléments au JCL de lancement de CICS ! JCL VSE !
! ! ! ! !
! 9INS ! Liste programmes et fichiers installés ! JCL VSE !
! 9TSG ! Tests des procédures du gestionnaire ! SYSIN !
! 9TSE ! Tests des utilitaires d'extraction ! SYSIN !
! 9TSU ! Tests du système TP ! texte !
! 9DSG ! Entités utilisateur PACDESIGN selon la ! SYSIN !
! variante : ! ! !
! ! MER = Méthode Merise ! ! !
! ! SSA = Méthode SSADM ! ! !
! ! YSM = Méthode YSM ! ! !
! ! IFW = Méthode IFW ! ! !
! ! OMT = Méthode OMT ! ! !
! 9DSP ! Paramétrage de PACDESIGN selon la ! SYSIN !
! variante : ! ! !
! ! MER = Méthode Merise ! ! !
! ! SSA = Méthode SSADM ! ! !
! ! YSM = Méthode YSM ! ! !
! ! IFW = Méthode IFW ! ! !
! ! OMT = Méthode OMT ! ! !
! 9INC ! Exemple cartes de contrôle optionnelles ! JCL VSE !
! 9PAF ! Source du dictionnaire PAF ! SYSIN !
! 9PQU ! Source des règles de qualité ! SYSIN !
! 9PQR ! Règles de qualité compilées ! fichier !
! 9XPD ! Source entité utilisateur (Ext PAF-PDM) ! SYSIN !
! ! ! ! !
-----
    
```

TABLEAU DES MODULES DE JCL D'EXPLOITATION :

```

-----
! Code ! Contenu !
!-----!-----!
!      !      !
! ACTI ! Analyse d'activité !
! ARCH ! Archivage du journal !
! CPSN ! Comparaison sous-réseaux !
! CRYP ! Cryptage/décryptage des mots de passe !
! CSES ! Compression des numéros de session !
! DCOB ! Edition COBOL documenté !
! EMLD ! Chargement des libellés d'erreur utilisateur !
! EMSN ! Extraction de sous-réseau pour fusion !
! EMUP ! Mise à jour des libellés d'erreur utilisateur !
! ESES ! Extraction des numéros de session !
! GETA ! Génération descriptifs de tables !
! GETD ! Génération descriptifs de tables env. disjointes !
! GETI ! Initialisation des fichiers de PACTABLES !
! GETO ! Initialisation des fichiers de PACTABLES !
! GET1 ! Génération descriptifs de tables !
! GET2 ! Génération descriptifs de tables !
! GPRT ! Edition-génération !
! GRPE ! Génération commandes de réorganisation de la base !
! HIPE ! Historisation automatique !
! IANA ! Analyse d'impact !
! IGRA ! Décomposition des zones groupes !
! INFO ! Initialisation FO !
! INFP ! Initialisation FP !
! INFQ ! Initialisation FQ !
! INPE ! Initialisation des fichiers PEI !
! IPEP ! Edition critères recherche !
! IPFQ ! Edition fichier FQ !
! IPIA ! Edition des résultats !
! ISEP ! Sélection points d'entrée !
! ISOS ! Sélection chaînes, opérateurs !
! LOAE ! Restauration des paramètres utilisateurs !
! LVBL ! Changement des low-values en space dans PC !
! MESN ! Fusion de sous-réseaux !
! MLIB ! Gestion des bibliothèques !
! PACX ! Extractions !
! PAFX ! Extracteur généralisé !
! PARM ! Mise à jour des paramètres utilisateurs !
! PQCA ! Pacbench Quality Control - analyse !
! PQCE ! Pacbench Quality Control - extraction !
-----

```

TABLEAU DES MODULES DE JCL D'EXPLOITATION (SUITE) :

```

-----
! Code ! Contenu !
!-----!-----!
!      !      !
! PRGS ! Edition du fichier des plans type !
! PRPE ! Editions environnements PEI !
! REAG ! Restauration des commandes d'édit-génération !
! REOR ! Réorganisation de la base !
! REST ! Restauration de la base !
! RESY ! Restauration système de la base !
! RMEN ! Normalisation !
! RMTD ! Restauration TD après migration !
! RSPE ! Restauration des fichiers PEI !
! RTLO ! Reprise des verrous !
! SADM ! Contrôle d'intégrité méthode SSADM !
! SASN ! Sauvegarde de sous-réseau !
! SASY ! Sauvegarde système de la base !
! SAVE ! Sauvegarde de la base !
! SIPE ! Simulation mise en production !
! SMTD ! Sauvegarde TD pour migration !
! STOP ! Optimisation stockage d'une base multi-volumes !
! SVAG ! Sauvegarde des commandes d'édit-génération !
! SVPE ! Sauvegarde des fichiers PEI !
! TCCI ! TeamConnection contrôle d'intégrité !
! TCGP ! TeamConnection valorisation des paramètres !
! TCLS ! TeamConnection mise à jour base !
! TRDU ! Production mouvements DSMS !
! TRJC ! Compression fichier journal !
! TRPF ! Génération mouvements transfert !
! TRRP ! Extraction du journal !
! TRUP ! Mise à jour paramètres !
! TRUV ! Transposition de manuels en rapports !
! UPDP ! Mise à jour de la base à partir des tables PAF !
! UPDT ! Mise à jour de la base !
! UXSR ! Extraction pour mise à jour sous-réseau !
! VDWN ! Descente VA Pac-VA Smalltalk !
! VINS ! Mise à jour entités utilisateur VA Pacbase !
! VPUR ! Purge entités VisualAge Pacbase !
! VUP1 ! Remontée codes VisualAge Pacbase !
! VUP2 ! Remontée mvts VisualAge Pacbase !
! XPAF ! Validation d'un plan-type d'extraction !
! XPDM ! Validation d'un plan-type d'édit !
! YSMC ! Contrôle d'intégrité méthode YSM !
-----

```

TABLEAU DES MODULES SOURCES POUR GENERATEURS CROISES :

! Code	! Contenu	!
! Z980	! Regroupe l'ensemble des modules suivants :	!
! ZARCVS	! MVS/CICS Cobol VS et VSE/CICS Cobol VS	!
! ZARCII	! MVS/CICS Cobol II	!
! ZARG7	! GCOS7/TDS	!
! ZARG8	! GCOS8/DMIV et TP8	!
! ZARICL	! ICL	!
! ZARBUR	! Unisys A	!
! ZARDEC	! DEC (caractères)	!
! ZARDE2	! DEC (champs)	!
! ZARTRM	! DEC (assembleur)	!
! SCRDEC	! sous-programme DEC	!
! HPFORM	! Traitement du message écran HP3000	!
! ZARMF1	! Microfocus	!
! SCRCODIF	! sous-programme Microfocus	!
! SCRIOPAR	! sous-programme Microfocus	!
! SCRPEINT	! sous-programme Microfocus	!
! SCRSAISI	! sous-programme Microfocus	!
! ZARMFO	! Réserve VisualAge Pacbase	!
! SCRMFO	! Réserve VisualAge Pacbase	!

4.9. PARAMETRAGE DU JCL

TABLEAUX DES PARAMETRES

Note : Si la valeur d'un paramètre contient des caractères spéciaux, elle doit être renseignée entre quotes (Ex. '1000,600').

PARAMETRAGE DES CARTES POWER :

```
-----  
!  CODE      ! SIGNIFICATION                                     !  
-----  
!           !  
! PRFJ      ! Préfixe des noms de Jobs POWER (4 caractères) !  
$ JOB      !  
$ JOB      !  
$ JOB      !  
$ JOB !  
!           ! ex : PWJ=' ,NTFY=USER1 '                       !  
$ LST      !  
$ LST      !  
$ LST !  
!           ! ex : PWL=' ,RBS=100,DEST=RMTNODE '             !  
$ PUN      !  
!           ! pour récupération sous éditeur.                 !  
$ PUN      !  
!           ! pour récupération sous éditeur.                 !  
$ PUN !  
!           ! pour récupération sous éditeur.                 !  
$ LST      !  
$ LST      !  
!           !  
-----
```

PARAMETRAGE DES NOMS DE FICHIERS VSAM :

Exceptés les fichiers de travail des tris, les fichiers VA Pac sont tous des fichiers VSAM dont les noms physiques sont de la forme :

```
PXU.PROPFiff      : fichiers VA Pac utilisateur
PXY.PROPROff      : fichiers VA Pac système
PXU.SEQ.PROPFI.OLDff : sauvegardes : génération (-1)
PXU.SEQ.PROPFI.ff   :                  génération (0)
PXU.SEQ.PROPFI.NEWff :                  génération (+1)
XW.WPROC.ff        : fichiers temporaires
PXU.RPROC.ff       : fichiers résultats de procédures
TXU.TRO00ff       : fichiers VA Pactables
```

```
-----
! CODE      ! SIGNIFICATION                                     !
-----
!           !
! PXU      ! Index des noms des fichiers VA Pac utilisateur !
! PXY      ! Index des noms des fichiers VA Pac système   !
! PRO      ! Radical du système VA Pac .                  !
!           !           (deux caractères différents de 99)   !
! PFI      ! Identifiant d'une base VA Pac .              !
!           !           (deux caractères différents de 99)   !
! XW       ! Index des noms de fichiers temporaires       !
!           ! (on doit le préfixer par '%' pour permettre  !
!           ! l'exécution simultanée de procédures dans  !
!           ! dans deux partitions ou plus : VSAM concatène !
!           ! l'identifiant de la partition au nom physique !
!           ! des fichiers comportant ce caractère en pos° 1)!
! VCAP     ! Nom du catalogue VSAM du système VA Pac      !
!           ! A l'installation, tous les fichiers sont    !
!           ! supposés sous le même catalogue.          !
!           !
----- ===SELV TAB (OPTION Pactables)-----
!           !
! TXU      ! Index des noms des fichiers VA Pactables     !
! TRO      ! Radical du système VA Pactables              !
!           !           (deux caractères différents de PW)   !
! VCAT     ! Nom du catalogue VSAM du système VA Pactables !
!           !
----- ===SELV DSM (OPTION DSMS) -----
!           !
! DSMS     ! Nom du fichier interface VA Pac -DSMS        !
! VCAD     ! Nom du catalogue VSAM du système DSMS        !
!           !
-----
```

PARAMETRAGE DES RESSOURCES UTILISEES (VOLUMES, DEROULEURS)

```

-----
! CODE ! SIGNIFICATION !
-----
! ! !
! TAPEI ! Nom de la copie de la bande d'installation. C'est !
! ! cette copie qui sera utilisée dans l'implantation !
! TAP1 ! Adresses (cuu) de deux dérouleurs !
! TAP2 ! ! !
! VOLVWK ! Nom de volume des fichiers VSAM temporaires !
! VOLPSV ! Nom de volume des fichiers de sauvegarde et !
! ! de journalisation. Il est judicieux pour garantir !
! ! la sécurité des données, de les mettre à part. !
! VOLPUD ! Nom de volume des composants DATA des fichiers !
! ! VA Pac utilisateur. !
! VOLPUX ! Nom de volume des composants INDEX des fichiers !
! ! VA Pac utilisateur. !
! VOLPYD ! Nom de volume des composants DATA des fichiers !
! ! VA Pac système. !
! VOLPYX ! Nom de volume des composants INDEX des fichiers !
! ! VA Pac système. !
! VOLTUD ! Nom de volume des composants DATA des fichiers !
! ! VA Pac TABLE utilisateur (si ==SELV TAB) !
! VOLTUX ! Nom de volume des composants INDEX des fichiers !
! ! VA Pac TABLE système (si ==SELV TAB) !
! VWORK ! Volume des fichiers de travail non-VSAM !
! WKBG1 ! 3 zones de travail sur ce volume sous la forme !
! WKBG2 ! WORK.='p,n' p=position de la lere piste !
! WKBG3 ! n=nombre de pistes allouées !
! WKFx1 ! Facultatif : équivalent des trois zones précédtes !
! WKFx2 ! si vous souhaitez permettre l'exécution concur- !
! WKFx3 ! rente de certaines procédures en BG, F2, F3... !
! RPCP ! Allocation primaire du fichier PC en records !
! RPCS ! Allocation secondaire du fichier PC en records !
! ! (Le fichier PC est la sauvegarde séquentielle !
! ! d'une base VA Pac , sur le volume VOLPSV) !
! RAJP ! Allocation primaire du fichier AJ en records !
! RAJS ! Allocation secondaire du fichier AJ en records !
! ! (Le fichier AJ est le fichier journal VA Pac , !
! ! sur le volume VOLPSV) !
-----

```

PARAMETRAGE DES LIBRAIRIES

```

-----
! CODE ! SIGNIFICATION !
-----
! ! !
! MODB ! Nom physique de la librairie (ou CIL) des !
! ! programmes batch du produit. !
! LMODB ! Nom de la librairie des programmes du produit. !
! SMODB ! Nom de la sous-librairie des programmes batch du !
! ! produit, précédé d'un point : SMODB='.sublib' !
! MODO ! Nom physique de la librairie (ou CIL) des !
! ! programmes TP. !
! SMODO ! Nom de la sous-librairie des programmes TP, !
! ! précédé d'un point : SMODO='.sublib' !
! PRLB ! Nom physique de la librairie (ou PL) des !
! ! procédures VSE. !
! LPRLB ! Nom de la librairie des procédures VSE. !
! SPRLB ! Nom de la sous-librairie des procédures VSE !
! ! précédé d'un point : SPRLB='.sublib' !
! SLIL ! Nom physique de la librairie (ou SL) des !
! ! membres SLI. !
! LSLI ! Nom de la librairie des membres SLI. !
! SLIS ! Nom de la sous-librairie des membres SLI !
! ! précédé d'un point : SLIS='.sublib' !
! SLIT ! Type des membres SLI. Par défaut, SLIT='PAC' !
-----

```

PARAMETRES DIVERS :

```

-----
! CODE ! SIGNIFICATION !
-----
! ! !
! OPT ! paramètres carte // OPTION !
! ! par défaut, OPT=LOG,NODUMP !
! SORTWK ! Nom de la procédure d'assignation des SORTWK !
! $SORTWK ! non VSAM, valeur par défaut !
! ! Les procédures $OSORTWK,$xSORTWK,... sont !
! ! fournies dans le module '1VSE' !
! ! SORTWK=... : SAM sous VSAM Space Management !
! ! La procédure SORTWK est fournie (module '1VSE') !
! ! SORTWK=... : Votre propre procédure... !
! LIST ! CICS/RDO (Resource definition online) : nom de !
! ! liste pour les ressources progiciel VA Pac !
! GROUP ! CICS/RDO (Resource definition online) : nom de !
! ! groupe pour les ressources du produit installé !
-----

```

4.10. ECLATEMENT DES MODULES DE JCL

LIGNES EN TETE OU EN FIN DE MODULES DE JCL

```
===BEGMOD
....1   )
.....   ) lignes à insérer avant chaque module de JCL
....n   )
===ENDMOD
....1   )
.....   ) lignes à insérer après chaque module de JCL
....n   )
```

Des lignes peuvent être insérées en entrée de MM1JCL si la valeur par défaut proposée ne convient pas (Voir au sous-chapitre "Valeurs par défaut installation" ci-dessus).

Ces lignes ont pour but d'effectuer la séparation du fichier de JCL créé par l'utilitaire MM1JCL en autant de membres différents que de modules de JCL.

L'utilitaire ajoute devant chaque module de JCL les lignes1 àn et en fin de chaque module de JCL les lignes1 àn.

Le paramètre symbolique W\$MODUL représente le nom du module.
Pour le concaténer avec une chaîne de caractères, insérez un point si cette chaîne se trouve immédiatement après.

Exemple : W\$MODUL.B donnera WXXXXB

```
W$MODUL..B donnera WXXXX.B
W$MODULB   ne sera pas résolu.
```

Quelques exemples vous sont présentés ci-dessous :

Pour automatiser le découpage en membres CMS/XEDIT :

1. Insérez les cartes suivantes :

```
===BEGMOD
:READ xxxx$MODUL (xxxx = préfixe de 4 car. maxi).
```

2. Exécutez le job de traitement du JCL.
3. Mettez le Punch reçu de POWER en tête du Reader CMS

(ORDER RDR nnnn où nnnn est le numéro du Spool).

4. Faites correspondre la classe du Reader et du Spool

(SPOOL RDR CLASS *).

5. Si vous désirez placer ces modules sur un disque partagé entre plusieurs machines, accédez celui-ci avec le filemode A

(ACCESS cuu A où cuu est l'adresse du disque).

6. Déclenchez le dispatching automatique en donnant le filetype désiré

(READ * filetype).

Note : il est possible de donner le filetype au niveau de la carte :READ insérée par la commande BEGMOD de la manière suivante :

```
:READ xxxxW$MODUL filetype
```

7. Restaurez la configuration initiale de votre machine CMS

(ACC 191 A puis ACC cuu filemode).

Pour automatiser le découpage en membres d'une librairie SP2 ou VSE/ESA :

1. Insérez les cartes :

```
===BEGMOD  
CA xxxx$MODUL..P (xxxx = préfixe de 4 car. maxi)  
===ENDMOD  
/+
```

2. Exécutez le job de traitement du JCL, en prévoyant le stockage du PUNCH de MM1JCL sur disque.

3. Reprenez le PUNCH en entrée de LIBR avec

```
ASSGN SYSIPT,DISK,.. (si le punch est sur disque)  
// EXEC LIBR,PARM=' ACCES SUBLIB=yourlib.sublib'
```

Pour automatiser le découpage en membres d'un PDS sous MVS:

1. Insérer les cartes:

```
===BEGMOD  
./ ADD NAME=$PRO$PFI$ZMODUL  
//%USER$MODUL JOB  
%ACCT,VSE,CLASS=X,NOTIFY=%USER  
/*XMIT nom_de_la_machine_VSE DLM=%  
===ENDMOD  
%%
```

Il s'agit d'ailleurs de l'exemple de JCL fourni en standard dans le step 2.

2. Exécutez le job de traitement du JCL.

3. Reprenez le PUNCH ainsi produit. On a chaque module de JCL séparé par la commande:

```
./ ADD NAME=<module_de_JCL>
```

Ce flot constituant l'entrée de IEBUPDTE servant à cataloguer ces différents modules dans le PDS prévu à cet effet.

4.11. DEROULEMENT DE L'IMPLANTATION

DEROULEMENT DE L'IMPLANTATION

Une fois les JCL obtenus, l'implantation du système VA Pac se déroule suivant les phases suivantes:

1. Mise à jour du JCL de POWER et lancement
2. Mise à jour des tables CICS
- 2.1. CSD
- 2.2. ASSEMBLEUR
3. Catalogage des procédures
- 3.1. Membres SLI
- 3.2. Procédures VSE
4. Installation du système VA Pac
- 4.1. Définition des fichiers permanents
- 4.2. Catalogage du JCL de GPRT (membre SLI)
- 4.3. Catalogage du JCL de PACX (membre SLI)
- 4.4. Définition des fichiers de travail de GPRT
- 4.5. Définition des fichiers de travail de PACX
- 4.6. Installation des modules exécutables.
- 4.7. Chargement des fichiers systèmes
- 4.8. Chargement des fichiers utilisateurs
- 4.9. Chargement des libellés d'erreur-documentation en ligne et des paramètres utilisateurs
5. Initialisation d'une base
- 5.1. Installation de la base de tests VisualAge Pacbase
- 5.2. Initialisation du fichier des commandes d'édition/génération
- 5.3. Initialisation des fichiers du Module Environnement de Production (variante PEI)
6. Compléments optionnels d'installation système VA Pac
- 6.1. Module VisualAge Smalltalk

6.2. Module PAF

6.2.1. Environnement

6.2.2. Extension PAF+

6.3. Module PACDESIGN

6.4. Module PQC (PACBENCH QUALITY CONTROL)

7. Modification du JCL de lancement de CICS

8. Autres compléments d'installation

8.1. Module Pac/Impact

8.2. Module PAC/TRANSFER

8.3. Module du Pont VA Pac - TeamConnection

9. Liste des programmes et fichiers de l'installation

4.12. 1. MODIFICATION DU JCL DE POWER

MODIFICATIONS DU JCL DE POWER

Le module de JCL 1PWR contient les lignes à ajouter dans le JCL de lancement de POWER pour pouvoir utiliser la fonction\$ SLI).

I N S T A L L A T I O N
1. MODIFICATION DU JCL DE POWER

```
===MOD 1PWR  
/* *****  
/*      --- MODIFICATIONS A APPORTER AU JCL DE POWER      *  
* =====  
*      AJOUTER $LSLI$SLIS  DANS LA  LIBDEF SOURCE,SEARCH  *  
* =====  
/* *****  
// DLBL $LSLI, '$SLIL', ,VSAM
```

4.13. 2.1 MISE A JOUR DES TABLES CICS (CSD)

MISE A JOUR DES TABLES CICS (CSD)

CICS/RDO: Les cartes de commandes de l'utilitaire DFHCSDUP dans le membre 1CSD, permettent la mise à jour de la DFHCSD.

La table des fichiers est mise à jour par le membre 1FCT (cf sous-chapitre suivant).

Les éléments contenus dans le membre 1CSD sont décrits dans le sous-chapitre suivant.

I N S T A L L A T I O N

4

2.1 MISE A JOUR DES TABLES CICS (CSD)

13

```

===MOD 1CSD
*****
***** $PROD $PREL PROGRAMS *****
*****
===SEQ FOR PUF
DEFINE PROGRAM($PRO.FANM) GROUP($GROUP)
DEFINE PROGRAM($PRO.FBIB) GROUP($GROUP)
DEFINE PROGRAM($PRO.FCTL) GROUP($GROUP)
DEFINE PROGRAM($PRO.FDBD) GROUP($GROUP)
DEFINE PROGRAM($PRO.FECR) GROUP($GROUP)
DEFINE PROGRAM($PRO.FENU) GROUP($GROUP)
DEFINE PROGRAM($PRO.FFOG) GROUP($GROUP)
DEFINE PROGRAM($PRO.FMCL) GROUP($GROUP)
DEFINE PROGRAM($PRO.FPGM) GROUP($GROUP)
DEFINE PROGRAM($PRO.FRUB) GROUP($GROUP)
DEFINE PROGRAM($PRO.FSDO) GROUP($GROUP)
DEFINE PROGRAM($PRO.FTXT) GROUP($GROUP)
DEFINE PROGRAM($PRO.FVER) GROUP($GROUP)
DEFINE PROGRAM($PRO.F000) GROUP($GROUP)
===SEQ
DEFINE PROGRAM($PRO.PA00) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA01) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA10) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA11) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA12) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA13) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA14) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA15) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA16) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA17) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA18) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA19) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA20) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA21) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA22) GROUP($GROUP)
===SEQ FOR PEI
DEFINE PROGRAM($PRO.PA30) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA31) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA32) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA33) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA34) GROUP($GROUP)
DEFINE PROGRAM($PRO.PA35) GROUP($GROUP)
===SEQ
DEFINE PROGRAM($PRO.PBND) GROUP($GROUP)
DEFINE PROGRAM($PRO.PPHP) GROUP($GROUP)
DEFINE PROGRAM($PRO.QA00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QB00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QC00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QC01) GROUP($GROUP)
DEFINE PROGRAM($PRO.QC50) GROUP($GROUP)
DEFINE PROGRAM($PRO.QD00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QE00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QF00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QF10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QG00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QH00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QH01) GROUP($GROUP)
DEFINE PROGRAM($PRO.QH20) GROUP($GROUP)
DEFINE PROGRAM($PRO.QH30) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI01) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI02) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI03) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI04) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI05) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI20) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI21) GROUP($GROUP)
DEFINE PROGRAM($PRO.QI50) GROUP($GROUP)
DEFINE PROGRAM($PRO.QK10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QK20) GROUP($GROUP)
DEFINE PROGRAM($PRO.QK30) GROUP($GROUP)
DEFINE PROGRAM($PRO.QL10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QL20) GROUP($GROUP)

```

I N S T A L L A T I O N

4

2.1 MISE A JOUR DES TABLES CICS (CSD)

13

```
DEFINE PROGRAM($PRO.QL21) GROUP($GROUP)
DEFINE PROGRAM($PRO.QL30) GROUP($GROUP)
===SEQ FOR SQL
DEFINE PROGRAM($PRO.QL40) GROUP($GROUP)
DEFINE PROGRAM($PRO.QL41) GROUP($GROUP)
DEFINE PROGRAM($PRO.QL45) GROUP($GROUP)
DEFINE PROGRAM($PRO.QL46) GROUP($GROUP)
===SEQ
DEFINE PROGRAM($PRO.QM00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP01) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP02) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP03) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP04) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP05) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP06) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP07) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP08) GROUP($GROUP)
DEFINE PROGRAM($PRO.QP50) GROUP($GROUP)
DEFINE PROGRAM($PRO.QR00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS02) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS03) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS04) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS05) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS06) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS08) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS09) GROUP($GROUP)
DEFINE PROGRAM($PRO.QS10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QT00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QT10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QT20) GROUP($GROUP)
DEFINE PROGRAM($PRO.QT50) GROUP($GROUP)
DEFINE PROGRAM($PRO.QUPA) GROUP($GROUP)
DEFINE PROGRAM($PRO.QUPF) GROUP($GROUP)
DEFINE PROGRAM($PRO.QU00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QU01) GROUP($GROUP)
DEFINE PROGRAM($PRO.QU10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QU20) GROUP($GROUP)
DEFINE PROGRAM($PRO.QV10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QV20) GROUP($GROUP)
DEFINE PROGRAM($PRO.QV30) GROUP($GROUP)
DEFINE PROGRAM($PRO.QX00) GROUP($GROUP)
DEFINE PROGRAM($PRO.QX01) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY01) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY02) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY03) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY04) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY05) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY10) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY11) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY20) GROUP($GROUP)
DEFINE PROGRAM($PRO.QY30) GROUP($GROUP)
DEFINE PROGRAM($PRO.QZ00) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q000) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q100) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q101) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q102) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q103) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q104) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q200) GROUP($GROUP)
===SEQ FOR TAB
DEFINE PROGRAM($PRO.Q210) GROUP($GROUP)
===SEQ
DEFINE PROGRAM($PRO.Q300) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q400) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q500) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q600) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q700) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q800) GROUP($GROUP)
DEFINE PROGRAM($PRO.Q900) GROUP($GROUP)
DEFINE PROGRAM($PRO.R000) GROUP($GROUP)
DEFINE PROGRAM($PRO.R005) GROUP($GROUP)
DEFINE PROGRAM($PRO.R100) GROUP($GROUP)
```

I N S T A L L A T I O N
2.1 MISE A JOUR DES TABLES CICS (CSD)

```
DEFINE PROGRAM($PRO.R200) GROUP($GROUP)
DEFINE PROGRAM($PRO.R400) GROUP($GROUP)
DEFINE PROGRAM($PRO.R500) GROUP($GROUP)
DEFINE PROGRAM($PRO.R600) GROUP($GROUP)
DEFINE PROGRAM($PRO.R980) GROUP($GROUP) RESIDENT(YES)
DEFINE PROGRAM($PRO.TPST) GROUP($GROUP)
DEFINE PROGRAM($PRO.TPWS) GROUP($GROUP)
DEFINE PROGRAM($PRO.UCTR) GROUP($GROUP) RESIDENT(YES)
*****
***** $PROD $PREL TRANSACTIONS *****
*****
DEFINE TRANSACTION($PRO$PFI) GROUP($GROUP) PROGRAM($PRO.R000)
DEFINE TRANSACTION($PRO.EF) GROUP($GROUP) PROGRAM($PRO.QUPF)
ADD GROUP($GROUP) LIST($LIST)
```

I N S T A L L A T I O N

4

2.1 MISE A JOUR DES TABLES CICS (CSD)

13

```

===MOD 1FCT
* *****
*
*   FCT CICS $PROD $PREL : $PRO$PFI.XX & $PRO$PRO.XX FILES
*
* *****
===SEQ FOR PEI
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AB,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( BROWSE ),                          $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=2,BUFNI=1,STRNO=1,                    $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AC,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
            LOG=YES,                                     $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
===SEQ
    DFHFCT TYPE=DATASET,DATASET=$PRO$PRO.AE,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
            LOG=YES,                                     $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AG,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=2,BUFNI=1,STRNO=1,                    $Y.*
            LOG=YES,                                     $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AJ,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=2,STRNO=1,LOG=YES,                     $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AN,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
            LOG=YES,                                     $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PRO.AP,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                    $Y.*
            LOG=YES,                                     $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AR,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*
            LSRPOOL=$NLSR,                               $Y.*
            BUFND=3,STRNO=2,                              $Y.*
            LOG=YES,                                     $Y.*
            RECFORM=( FIXED,BLOCKED ),                  $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PRO.PA,          $Y.*
            ACCMETH=VSAM,                              $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),      $Y.*

```

I N S T A L L A T I O N
2.1 MISE A JOUR DES TABLES CICS (CSD)

PAGE

94

4
13

LSRPOOL=\$NLSR,
BUFND=2, BUFNI=1, STRNO=1,
RECFORM=(VARIABLE, BLOCKED),
FILSTAT=(ENABLED, CLOSED)

\$Y.*
\$Y.*
\$Y.*

4.14. 2.2 MISE A JOUR DES TABLES CICS (ASSEMBLEUR)

MISE A JOUR DES TABLES CICS (ASSEMBLEUR)

Sources assembleur:

PCT: Module de JCL '1PCT' - 2 codes transaction:

\$PRO\$PFI (Option DTB=YES obligatoire), pour l'accès à VA PAC .

Une transaction par base installée.

\$PRO.EF (Option DTB=YES obligatoire), pour la gestion des paramètres utilisateur et le module environnement de production.

Une transaction pour le système VA PAC .

PPT: Module de JCL '1PPT' - Maps et Programmes codifiés :

\$PRO.Qxxx, \$PRO.Pxxx ou \$PRO.Rxxx pour les programmes
\$PRO.Nxxx, \$PRO.Mxxx ou \$PRO.Oxxx pour les maps,
xxx étant un code alphanumérique.
(Voir la liste complète au Sous-chapitre 'Les bibliothèques' au
Chapitre 'INTRODUCTION').

Une PPT pour le système VA PAC .

FCT: Module de JCL '1FCT':

2 fichiers communs au système VA PAC :

\$PRO\$PRO.AE : Libellés d'erreurs
\$PRO\$PRO.AP : Paramètres utilisateur

4 fichiers utilisateur par base installée :

\$PRO\$PFI.AN : Base VA PAC , fichier des index,
\$PRO\$PFI.AR : Base VA PAC , fichier des données,
\$PRO\$PFI.AJ : Journal des mouvements,
\$PRO\$PFI.AG : Commandes d'édition et génération.

2 fichiers utilisateur du module PEI par base installée:

\$PRO\$PFI.AB : Environnement de production consulté
\$PRO\$PFI.AC : Environnement de production mis à jour

1 fichier pour le module TABLES, commun au système :

\$TRO.00TD : Descriptif des tables.

1 fichier de travail pour PACSQL, commun au système :

\$PRO\$PRO.PA

L'option LOG=YES pour la FCT est obligatoire pour les fichiers AN, AR, AJ, AG, AP et pour AC du module PEI.

Le nombre de STRING, de BUFFER d'INDEX et de BUFFER de DATA prévus sont les nombres minimum nécessaires pour le système.

Les deux caractères '\$PRO' des codes transactions, des noms des maps et des programmes et des codes fichiers sont choisis par l'utilisateur.

La transaction \$PRO\$PFI permet d'effectuer les opérations courantes sur la base (consultation, mise à jour).

La transaction \$PRO.EF permet d'effectuer les opérations courantes sur les paramètres utilisateur gérés dans les fichiers AE et AP, ainsi que les opérations du Module Environnement de Production dans les fichiers AB et AC.

I N S T A L L A T I O N
2.2 MISE A JOUR DES TABLES CICS (ASSEMBLEUR)

```
==MOD 1PCT
* *****
*
*   PCT CICS $PROD $PREL : $PRO.XX TRANSACTIONS
*
* *****
*   DFHPCT TYPE=ENTRY,TRANSID=$PRO.$PFI.,           $Y.*
*           PROGRAM=$PRO.R000,DTB=YES
*   DFHPCT TYPE=ENTRY,TRANSID=$PRO.EF,             $Y.*
*           PROGRAM=$PRO.QUPF,DTB=YES
* LE POSTE CI-DESSOUS EST LA VERSION ANGLAISE DE $PRO.EF
*   DFHPCT TYPE=ENTRY,TRANSID=$PRO.EE,           $Y.*
*           PROGRAM=$PRO.QUPA,DTB=YES
```

I N S T A L L A T I O N

4

2.2 MISE A JOUR DES TABLES CICS (ASSEMBLEUR)

14

```

==MOD 1PPT
* *****
*
*   PPT CICS $PROD $PREL : PROGRAMS $PRO.XXXX
*
* *****

==SEQ FOR PUF
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FANM, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FBIB, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FCTL, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FDBD, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FECC, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FENU, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FFOG, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FMCL, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FPGM, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FRUB, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FSDO, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FTXT, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.FVER, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.F000, PGMLANG=COBOL

==SEQ
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA01, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA10, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA11, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA12, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA13, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA14, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA15, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA16, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA17, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA18, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA19, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA20, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA21, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA22, PGMLANG=COBOL

==SEQ FOR PEI
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA30, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA31, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA32, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA33, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA34, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PA35, PGMLANG=COBOL

==SEQ
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PBND, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.PPHP, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QA00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QB00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QC00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QC01, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QC50, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QD00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QE00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QF00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QF10, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QG00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QH00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QH01, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QH20, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QH30, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI00, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI01, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI02, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI03, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI04, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI05, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI20, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI21, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QI50, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QK10, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QK20, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=$PRO.QK30, PGMLANG=COBOL

```


I N S T A L L A T I O N
2.2 MISE A JOUR DES TABLES CICS (ASSEMBLEUR)

PAGE

100

4

14

DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.R005, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.R100, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.R200, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.R400, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.R500, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.R600, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.R980, PGMLANG=COBOL, RES=YES
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.TPST, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.TPWS, PGMLANG=COBOL
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.UCTA, PGMLANG=COBOL, RES=YES
DFHPPT TYPE=ENTRY, PROGRAM=\$PRO.UCTR, PGMLANG=COBOL, RES=YES

I N S T A L L A T I O N

4

2.2 MISE A JOUR DES TABLES CICS (ASSEMBLEUR)

14

```

===MOD 1FCT
* *****
*
*   FCT CICS $PROD $PREL : $PRO$PFI.XX & $PRO$PRO.XX FILES
*
* *****
===SEQ FOR PEI
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AB,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( BROWSE ),                       $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=2,BUFNI=1,STRNO=1,                 $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AC,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                 $Y.*
            LOG=YES,                                  $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
===SEQ
    DFHFCT TYPE=DATASET,DATASET=$PRO$PRO.AE,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                 $Y.*
            LOG=YES,                                  $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AG,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=2,BUFNI=1,STRNO=1,                 $Y.*
            LOG=YES,                                  $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AJ,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=2,STRNO=1,LOG=YES,                 $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AN,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                 $Y.*
            LOG=YES,                                  $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PRO.AP,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=3,BUFNI=2,STRNO=2,                 $Y.*
            LOG=YES,                                  $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PFI.AR,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*
            LSRPOOL=$NLSR,                            $Y.*
            BUFND=3,STRNO=2,                          $Y.*
            LOG=YES,                                  $Y.*
            RECFORM=( FIXED,BLOCKED ),               $Y.*
            FILSTAT=( ENABLED,OPENED )
    DFHFCT TYPE=DATASET,DATASET=$PRO$PRO.PA,          $Y.*
            ACCMETH=VSAM,                             $Y.*
            SERVREQ=( ADD,UPDATE,DELETE,BROWSE ),    $Y.*

```

I N S T A L L A T I O N
2.2 MISE A JOUR DES TABLES CICS (ASSEMBLEUR)

PAGE

102

4
14

LSRPOOL=\$NLSR,
BUFND=2, BUFNI=1, STRNO=1,
RECFORM=(VARIABLE, BLOCKED),
FILSTAT=(ENABLED, CLOSED)

\$Y.*
\$Y.*
\$Y.*

4.15. 3.1 PROCEDURES SLI A CATALOGUER

\$ SLI in POWER)

Le module de JCL 1SLI contient les sources de procédures à cataloguer dans une bibliothèque à laquelle on peut accéder par POWER en lecture (carte //LIBDEF SOURCE,SEARCH dans le JCL de lancement de POWER).

Ces procédures sont :

- DBJB\$PRO.\$PFI début de job standard
- JPAF\$PRO.\$PFI JCL standard programme utilisateur PAF
- DF\$PRO.\$PFI.ff Delete-define du fichier utilisateur ff
- DF\$PRO.\$PRO.ff Delete-define du fichier système ff
- D\$PRO.\$PRO.fff Delete-define du fichier fff
- DFSYSPAF Delete-define du fichier SYSPAF
 (utilisateurs de PACSQL)
- RG\$PRO.\$PFI.ff Rotation du fichier de sauvegarde ff

Les membres DFxxxxff et Dxxxxfff permettent de gérer de manière centralisée les paramètres d'allocation des fichiers VSAM de l'environnement VA PAC .

Les membres RGxxxxff permettent de gérer de manière centralisée le nombre de générations des fichiers de sauvegarde que vous désirez conserver.

Le membre DBJBxxxx permet de gérer de manière centralisée le JCL commun à toutes les procédures d'exploitation VA PAC (allocation de ressources telles que bibliothèques) se trouvant en début de job.

I N S T A L L A T I O N

4

3.1 PROCEDURES SLI A CATALOGUER

15

```

===MOD 1SLI
* $X JOB JNM=$PRFJ.1SLI,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.1SLI          ***** $PROD $PREL *****
/* CATALOGAGE DES PROCEDURES
/* *****
// OPTION $OPT
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL $LSLI,'$SLIL',,VSAM
/* GOTO JCLSTXX
* ----- STEP 1 ----- LIBR
// EXEC LIBR,PARM=' AC S=$LSLI$SLIS'
CA DBJB$PRO.$PFI..$SLIT R=Y
// ASSGN SYS005,SYSLST
// ASSGN SYS006,SYSRDR
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL $LMODEB,'$MODB',,VSAM
// DLBL $LPRLB,'$PRLB',,VSAM
// LIBDEF PHASE,SEARCH=$LMODEB$SMODEB
// LIBDEF PROC,SEARCH=$LPRLB$SPRLB
/+
CA PAFB$PRO.$PFI..$SLIT R=Y
* --- STEP 1 --- PAFP10
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO$PFI.AR',,VSAM
// EXEC PROC=$$PACWK1,DLBL=COB80,ASSGN=SYS015
// EXEC PAFP10,SIZE=AUTO
CBL APOST,LIB,NOADV,TRUNC(BIN),NOXREF,RES,RENT
* $X DATA INPUT
* $X /*
* --- STEP 2 --- IGYCRCTL
// EXEC PROC=$$PACWK1,ASSGN=SYSIPT
// EXEC IGYCRCTL,SIZE=IGYCRCTL
* --- STEP 3 --- LNKEDT
// EXEC LNKEDT
* $X /&
CLOSE SYSIPT,SYSRDR
/+
CA PAFT$PRO.$PFI..$SLIT R=Y
* --- STEP 1 --- PAFP10
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO$PFI.AR',,VSAM
// EXEC PROC=$$PACWK1,DLBL=COB80,ASSGN=SYS015
// EXEC PAFP10,SIZE=AUTO
CBL APOST,LIB,NOADV,TRUNC(BIN),NOXREF,RES,RENT
CBL XOPTS(CICS,OPT,NOLIST)
* $X DATA INPUT
* $X /*
* --- STEP 2 --- DFHECP1$
// ASSGN SYS091,SYSPCH
// EXEC PROC=$$PACWK1,ASSGN=SYSIPT
// EXEC PROC=$$PACWK2,ASSGN=SYSPCH
// EXEC DFHECP1$
* --- STEP 3 --- IGYCRCTL
CLOSE SYSIPT,SYSRDR
CLOSE SYSPCH,SYS091
// EXEC PROC=$$PACWK2,ASSGN=SYSIPT
INCLUDE DFHECI
// EXEC IGYCRCTL,SIZE=IGYCRCTL
* --- STEP 4 --- LNKEDT
// EXEC LNKEDT
* $X /&
CLOSE SYSIPT,SYSRDR
/+
CA JPAF$PRO.$PFI..$SLIT R=Y
/* == TOP OF PAF JCL SKELETON ==
// DLBL $PRO$PRO.CT,'$VCAP',,VSAM
// DLBL SY8PAF,'$XW..SYSPAF',,VSAM,CAT=$PRO$PRO.CT

```

INSTALLATION
3.1 PROCEDURES SLI A CATALOGUER

PAGE

105

4
15

```
// DLBL PAC7AE, '$PXY..$PRO$PRO.AE', ,VSAM,CAT=$PRO$PRO.CT
// DLBL PAC7AN, '$PXU..$PRO$PFI.AN', ,VSAM,CAT=$PRO$PRO.CT
// DLBL PAC7AR, '$PXU..$PRO$PFI.AR', ,VSAM,CAT=$PRO$PRO.CT
/* == END OF PAF JCL SKELETON ==
/+
CA DFMDFBK.$SLIT R=Y
/* ----- DELETE-DEFINE FIXBLK MODEL ----- */
DEL          ($MODEL F) CL
DEF CL       (NAME ($MODEL F) -
              VOL($VOLVWK) -
              REC (100 100) -
              RECSZ(80 80) -
              CISZ (8192) -
              RECFM(FB(80)) -
              RUS -
              NAL -
              NIXD -
              ) DATA (NAME ($MODEL F..D) )
/+
CA DFMDFBK.$SLIT R=Y
/* ----- DELETE-DEFINE VARBLK MODEL ----- */
DEL          ($MODEL V) CL
DEF CL       (NAME ($MODEL V) -
              VOL($VOLVWK) -
              REC (100 100) -
              RECSZ(80 200) -
              CISZ (8192) -
              RECFM(VB) -
              RUS -
              NAL -
              NIXD -
              ) DATA (NAME ($MODEL V..D) )
/+
CA DF$PRO.$PFI.AB.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AB ----- */
DEL          ($PXU..$PRO.$PFI.AB) CL
DEF CL       (NAME ($PXU..$PRO.$PFI.AB) -
              SHR(4) RUS KEYS(26 0) -
              RECSZ(110 110) REC(200 200) -
              ) INDEX (NAME ($PXU..$PRO.$PFI.AB.I) -
                       VOL($VOLPUX) -
              ) DATA (NAME ($PXU..$PRO.$PFI.AB.D) -
                       VOL($VOLPUD) FSPC(10 5) CISZ(4096) )
/+
CA DF$PRO.$PFI.AC.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AC ----- */
DEL          ($PXU..$PRO.$PFI.AC) CL
DEF CL       (NAME ($PXU..$PRO.$PFI.AC) -
              SHR(2 3) RUS KEYS(26 0) -
              RECSZ(110 110) REC(200 200) -
              ) INDEX (NAME ($PXU..$PRO.$PFI.AC.I) -
                       VOL($VOLPUX) -
              ) DATA (NAME ($PXU..$PRO.$PFI.AC.D) -
                       VOL($VOLPUD) FSPC (10 5) CISZ(4096) )
/+
CA DF$PRO.$PRO.AE.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AE ----- */
DEL          ($PXY..$PRO.$PRO.AE) CL
DEF CL       (NAME ($PXY..$PRO.$PRO.AE) -
              SHR(2 3) RUS KEYS(12 0) -
              RECSZ(80 80) REC(30000 1000) -
              ) INDEX (NAME ($PXY..$PRO.$PRO.AE.I) -
                       VOL($VOLPYX) -
              ) DATA (NAME ($PXY..$PRO.$PRO.AE.D) -
                       VOL($VOLPYD) FSPC(2 1) CISZ(4096) )
/+
CA DF$PRO.$PFI.AG.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AG ----- */
DEL          ($PXU..$PRO.$PFI.AG) CL
DEF CL       (NAME ($PXU..$PRO.$PFI.AG) -
              SHR(2) RUS KEYS(27 0) -
              RECSZ(150 150) REC(500 500) -
              ) INDEX (NAME ($PXU..$PRO.$PFI.AG.I) -
```

I N S T A L L A T I O N

4

3.1 PROCEDURES SLI A CATALOGUER

15

```

        VOL($VOLPUX) -
    ) DATA (NAME ($PXU..$PRO.$PFI.AG.D) -
            VOL($VOLPUD) FSPC(10 5) CISZ(4096) )
/+
CA DF$PRO.$PFI.AJ.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AJ ----- */
DEL          ($PXU..$PRO.$PFI.AJ) CL
DEF CL      (NAME ($PXU..$PRO.$PFI.AJ) -
            SHR(2) RUS NUMD VOL($VOLPUD) -
            RECSZ(167 167) REC($RAJP $RAJS) -
            ) DATA (NAME ($PXU..$PRO.$PFI.AJ.D) -
                    CISZ(8192) )
/+
CA DF$PRO.$PFI.AN.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AN ----- */
DEL          ($PXU..$PRO.$PFI.AN) CL
DEF CL      (NAME ($PXU..$PRO.$PFI.AN) -
            SHR(2) RUS KEYS(43 0) -
            RECSZ(54 54) REC(9000 2000) -
            ) INDEX (NAME ($PXU..$PRO.$PFI.AN.I) -
                    VOL($VOLPUX) -
            ) DATA (NAME ($PXU..$PRO.$PFI.AN.D) -
                    VOL($VOLPUD) FSPC(5 5) -
                    CISZ(8192) BUFSP(35840) )
/+
CA DF$PRO.$PRO.AP.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AP ----- */
DEL          ($PXU..$PRO.$PRO.AP) CL
DEF CL      (NAME ($PXU..$PRO.$PRO.AP) -
            SHR(2 3) RUS KEYS(7 0) -
            RECSZ(80 80) REC(500 500) -
            ) INDEX (NAME ($PXU..$PRO.$PRO.AP.I) -
                    VOL($VOLPYX) -
            ) DATA (NAME ($PXU..$PRO.$PRO.AP.D) -
                    VOL($VOLPYD) FSPC(10 5) CISZ(4096) )
/+
CA DF$PRO.$PFI.AR.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AR ----- */
DEL          ($PXU..$PRO.$PFI.AR) CL
DEF CL      (NAME ($PXU..$PRO.$PFI.AR) -
            SHR(2) RUS NUMD VOL($VOLPUD) -
            RECSZ(140 140) REC(40000 15000) -
            ) DATA (NAME ($PXU..$PRO.$PFI.AR.D) -
                    CISZ(8192) )
/+
CA D$PRO.$PRO.AE0.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER AE0 ----- */
DEL          ($PXU..$SEQ.$PRO$PRO..AE0) CL
DEF CL      (NAME ($PXU..$SEQ.$PRO$PRO..AE0) -
            MODEL($MODEL) -
            VOL($VOLPSV) REC(30000 1000) -
            ) DATA (NAME ($PXU..$SEQ.$PRO$PRO..AE0.D) )
/+
CA DF$PRO.$PRO.GS.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER GS ----- */
DEL          ($PXU..$PRO.$PRO.GS) CL
DEF CL      (NAME ($PXU..$PRO.$PRO.GS) -
            SHR(2 3) RUS KEYS(25 0) -
            RECSZ(203 203) REC(1000 1000) -
            ) INDEX (NAME ($PXU..$PRO.$PRO.GS.I) -
                    VOL($VOLPUX) -
            ) DATA (NAME ($PXU..$PRO.$PRO.GS.D) -
                    VOL($VOLPUD) FSPC (5 5) CISZ(4096) )
/+
CA DF$PRO.$PRO.PA.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER PA ----- */
DEL          ($PXU..$PRO.$PRO.PA) CL
DEF CL      (NAME ($PXU..$PRO.$PRO.PA) -
            SHR (2 3) RUS KEYS (37 2) -
            RECSZ(200 539) REC(10000 10000) -
            ) INDEX (NAME ($PXU..$PRO.$PRO.PA.I) -
                    VOL($VOLPYX) -
            ) DATA (NAME ($PXU..$PRO.$PRO.PA.D) -

```

```

                                VOL($VOLPYD) FSPC(5 5) CISZ(8192) )
/+
CA DF$PRO.$PFI.PC.$SLIT R=Y
/* ----- DELETE-DEFINE FICHIER PC ----- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPC) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWPC) -
            MODEL($MODELV) -
            RECSZ(63 159) REC($RPCP $RPCS) -
            VOL($VOLPSV) -
            ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPC.D) )
/+
CA DF$PRO.$PFI.PD.$SLIT R=Y
/* ----- DELETE-DEFINE FICHIER PD ----- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPD) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWPD) -
            MODEL($MODELV) -
            RECSZ(63 159) REC($RPCP $RPCS) -
            VOL($VOLPSV) -
            ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPD.D) )
/+
CA DF$PRO.$PRO.SC.$SLIT R=Y
/* ----- DELETE-DEFINE FICHIER SC ----- */
DEL          ($PXY..$PRO.$PRO.SC) CL
DEF CL      (NAME ($PXY..$PRO.$PRO.SC) -
            SHR(2 3) RUS KEYS(4 0) -
            ) INDEX (NAME ($PXY..$PRO.$PRO.SC.I) -
                    VOL($VOLPYX) -
                    ) DATA (NAME ($PXY..$PRO.$PRO.SC.D) -
                              RECSZ(3204 3204) CISZ(22528) -
                              VOL($VOLPYD) REC (40 5) )
/+
CA DF$PRO.$PRO.SF.$SLIT R=Y
/* ----- DELETE-DEFINE FICHIER SF ----- */
DEL          ($PXY..$PRO.$PRO.SF) CL
DEF CL      (NAME ($PXY..$PRO.$PRO.SF) -
            MODEL($MODELV) -
            SHR(2 3) -
            RECFM(FB(119)) -
            RECSZ(119 119) REC(3000 100) -
            VOL($VOLPYD) -
            ) DATA (NAME ($PXY..$PRO.$PRO.SF.D) )
/+
CA DF$PRO.$PRO.SG.$SLIT R=Y
/* ----- DELETE-DEFINE FICHIER SG ----- */
DEL          ($PXY..$PRO.$PRO.SG) CL
DEF CL      (NAME ($PXY..$PRO.$PRO.SG) -
            SHR(2 3) RUS KEYS(5 0) -
            ) INDEX (NAME ($PXY..$PRO.$PRO.SG.I) -
                    VOL($VOLPYX) -
                    ) DATA (NAME ($PXY..$PRO.$PRO.SG.D) -
                              RECSZ(4605 4605) CISZ(22528) -
                              VOL($VOLPYD) REC(400 10) )
/+
CA DF$PRO.$PRO.SP.$SLIT R=Y
/* ----- DELETE-DEFINE FICHIER SP ----- */
DEL          ($PXY..$PRO.$PRO.SP) CL
DEF CL      (NAME ($PXY..$PRO.$PRO.SP) -
            SHR(2 3) RUS KEYS(5 0) -
            ) INDEX (NAME ($PXY..$PRO.$PRO.SP.I) -
                    VOL($VOLPYX) -
                    ) DATA (NAME ($PXY..$PRO.$PRO.SP.D) -
                              RECSZ(4605 4605) CISZ(22528) -
                              VOL($VOLPYD) REC(5 1) )
/+
CA DF$PRO.$PRO.SR.$SLIT R=Y
/* ----- DELETE-DEFINE FICHIER SR ----- */
DEL          ($PXY..$PRO.$PRO.SR) CL
DEF CL      (NAME ($PXY..$PRO.$PRO.SR) -
            SHR (2 3) RUS KEYS(5 0) -
            ) INDEX (NAME ($PXY..$PRO.$PRO.SR.I) -
                    VOL($VOLPYX) -
                    ) DATA (NAME ($PXY..$PRO.$PRO.SR.D) -
                              RECSZ(4605 4605) CISZ(22528) -
```

```

                                VOL($VOLPYD) REC(30 5) )
/+
CA DF$PRO.$PRO.SS.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER SS ----- */
DEL          ($PX..$PRO.$PRO.SS) CL
DEF CL      (NAME ($PX..$PRO.$PRO.SS) -
             SHR(2 3) RUS KEYS(5 0) -
             ) INDEX (NAME ($PX..$PRO.$PRO.SS.I) -
                     VOL($VOLPYX) -
             ) DATA (NAME ($PX..$PRO.$PRO.SS.D) -
                     RECSZ(4605 4605) CISZ(22528) -
                     VOL($VOLPYD) REC(300 10) )
/+
CA DFSYSPAF.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER SYSPAF ----- */
DEL          ($XW..SYSPAF) CL
DEF CL      (NAME ($XW..SYSPAF) -
             SHR(2 3) RUS NAL KEYS(12 0) -
             RECSZ(170 468) REC(1000 1000) -
             VOL($VOLVWK) -
             ) INDEX (NAME ($XW..SYSPAF.I) -
             ) DATA (NAME ($XW..SYSPAF.D) CISZ(8192) )
/+
CA D$PRO.$PRO.TCO.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER TCO ----- */
DEL          ($PX..$PRO$PRO.TCO) CL
DEF CL      (NAME ($PX..$PRO$PRO.TCO) -
             MODEL($MODEL$) -
             VOL($VOLPSV) REC(50 10) -
             RECSZ(117 117) RECFM(FB(117)) -
             ) DATA (NAME ($PX..$PRO$PRO.TCO.D) )
/+
CA DF$TRO.00TD.$SLIT R=Y
/* --- DELETE-DEFINE FICHER 00TD --- */
DEL          ($TXU..$TRO.00TD) CL
DEF CL      (NAME ($TXU..$TRO.00TD) -
             SHR(2 3) RUS KEYS(21 0) -
             RECSZ(240 240) REC(1000 1000) -
             ) INDEX (NAME ($TXU..$TRO.00TD.I) -
                     VOL($VOLTUX) -
             ) DATA (NAME ($TXU..$TRO.00TD.D) -
                     VOL($VOLTUD) FSPC(10 5) CISZ(4096) )
/+
CA D$PRO.$PRO.VGE.$SLIT R=Y
/* ----- DELETE-DEFINE FICHER VGE ----- */
DEL          ($PX..$PRO$PRO.VGE) CL
DEF CL      (NAME ($PX..$PRO$PRO.VGE) -
             MODEL($MODEL$) -
             VOL($VOLPSV) REC(100 10) -
             RECSZ(117 117) RECFM(FB(117)) -
             ) DATA (NAME ($PX..$PRO$PRO.VGE.D) )
/+
CA RG$PRO.$PFI.FH.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE FH --- */
DEL          ($PXU..SEQ.$PRO$PFI..OLDFH) CL
ALTER       $PXU..SEQ.$PRO$PFI..FH -
NEWNAME     ($PXU..SEQ.$PRO$PFI..OLDFH)
ALTER       $PXU..SEQ.$PRO$PFI..FH.D -
NEWNAME     ($PXU..SEQ.$PRO$PFI..OLDFH.D)
ALTER       $PXU..SEQ.$PRO$PFI..NEWFH -
NEWNAME     ($PXU..SEQ.$PRO$PFI..FH)
ALTER       $PXU..SEQ.$PRO$PFI..NEWFH.D -
NEWNAME     ($PXU..SEQ.$PRO$PFI..FH.D)
/+
CA RG$PRO.$PFI.FO.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE FO --- */
DEL          ($PXU..SEQ.$PRO$PFI..OLDFO) CL
ALTER       $PXU..SEQ.$PRO$PFI..FO -
NEWNAME     ($PXU..SEQ.$PRO$PFI..OLDFO)
ALTER       $PXU..SEQ.$PRO$PFI..FO.D -
NEWNAME     ($PXU..SEQ.$PRO$PFI..OLDFO.D)
ALTER       $PXU..SEQ.$PRO$PFI..NEWFO -
NEWNAME     ($PXU..SEQ.$PRO$PFI..FO)

```

I N S T A L L A T I O N

4

3.1 PROCEDURES SLI A CATALOGUER

15

```

ALTER          $PXU..SEQ.$PRO$PFI..NEWFO.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..FO.D)
/+
CA RG$PRO.$PFI.FQ.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE FQ --- */
DEL            ($PXU..SEQ.$PRO$PFI..OLDFQ) CL
ALTER          $PXU..SEQ.$PRO$PFI..FQ -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDFQ)
ALTER          $PXU..SEQ.$PRO$PFI..FQ.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDFQ.D)
ALTER          $PXU..SEQ.$PRO$PFI..NEWFQ -
NEWNAME        ($PXU..SEQ.$PRO$PFI..FQ)
ALTER          $PXU..SEQ.$PRO$PFI..NEWFQ.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..FQ.D)
/+
CA RG$PRO.$PFI.FR.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE FR --- */
DEL            ($PXU..SEQ.$PRO$PFI..OLDFR) CL
ALTER          $PXU..SEQ.$PRO$PFI..FR -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDFR)
ALTER          $PXU..SEQ.$PRO$PFI..FR.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDFR.D)
ALTER          $PXU..SEQ.$PRO$PFI..NEWFR -
NEWNAME        ($PXU..SEQ.$PRO$PFI..FR)
ALTER          $PXU..SEQ.$PRO$PFI..NEWFR.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..FR.D)
/+
CA RG$PRO.$PFI.PC.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE PC --- */
DEL            ($PXU..SEQ.$PRO$PFI..OLDPC) CL
ALTER          $PXU..SEQ.$PRO$PFI..PC -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDPC)
ALTER          $PXU..SEQ.$PRO$PFI..PC.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDPC.D)
ALTER          $PXU..SEQ.$PRO$PFI..NEWPC -
NEWNAME        ($PXU..SEQ.$PRO$PFI..PC)
ALTER          $PXU..SEQ.$PRO$PFI..NEWPC.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..PC.D)
/+
CA RG$PRO.$PFI.PD.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE PD --- */
DEL            ($PXU..SEQ.$PRO$PFI..OLDPD) CL
ALTER          $PXU..SEQ.$PRO$PFI..PD -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDPD)
ALTER          $PXU..SEQ.$PRO$PFI..PD.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDPD.D)
ALTER          $PXU..SEQ.$PRO$PFI..NEWPD -
NEWNAME        ($PXU..SEQ.$PRO$PFI..PD)
ALTER          $PXU..SEQ.$PRO$PFI..NEWPD.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..PD.D)
/+
CA RG$PRO.$PRO.PE.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE PE --- */
DEL            ($PXU..SEQ.$PRO$PRO..OLDPE) CL
ALTER          $PXU..SEQ.$PRO$PRO..PE -
NEWNAME        ($PXU..SEQ.$PRO$PRO..OLDPE)
ALTER          $PXU..SEQ.$PRO$PRO..PE.D -
NEWNAME        ($PXU..SEQ.$PRO$PRO..OLDPE.D)
ALTER          $PXU..SEQ.$PRO$PRO..NEWPE -
NEWNAME        ($PXU..SEQ.$PRO$PRO..PE)
ALTER          $PXU..SEQ.$PRO$PRO..NEWPE.D -
NEWNAME        ($PXU..SEQ.$PRO$PRO..PE.D)
/+
CA RG$PRO.$PFI.PG.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE PG --- */
DEL            ($PXU..SEQ.$PRO$PFI..OLDPG) CL
ALTER          $PXU..SEQ.$PRO$PFI..PG -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDPG)
ALTER          $PXU..SEQ.$PRO$PFI..PG.D -
NEWNAME        ($PXU..SEQ.$PRO$PFI..OLDPG.D)
ALTER          $PXU..SEQ.$PRO$PFI..NEWPG -
NEWNAME        ($PXU..SEQ.$PRO$PFI..PG)
ALTER          $PXU..SEQ.$PRO$PFI..NEWPG.D -

```

I N S T A L L A T I O N
3.1 PROCEDURES SLI A CATALOGUER

PAGE

110

4
15

```
NEWNAME          ($PXU..SEQ.$PRO$PFI..PG.D)
/+
CA RG$PRO.$PFI.PJ.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE PJ --- */
DEL              ($PXU..SEQ.$PRO$PFI..OLDPJ) CL
ALTER           $PXU..SEQ.$PRO$PFI..PJ -
NEWNAME         ($PXU..SEQ.$PRO$PFI..OLDPJ)
ALTER           $PXU..SEQ.$PRO$PFI..PJ.D -
NEWNAME         ($PXU..SEQ.$PRO$PFI..OLDPJ.D)
ALTER           $PXU..SEQ.$PRO$PFI..NEWPJ -
NEWNAME         ($PXU..SEQ.$PRO$PFI..PJ)
ALTER           $PXU..SEQ.$PRO$PFI..NEWPJ.D -
NEWNAME         ($PXU..SEQ.$PRO$PFI..PJ.D)
/+
CA RG$PRO.$PFI.PP.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE PP --- */
DEL              ($PXU..SEQ.$PRO$PFI..OLDPP) CL
ALTER           $PXU..SEQ.$PRO$PFI..PP -
NEWNAME         ($PXU..SEQ.$PRO$PFI..OLDPP)
ALTER           $PXU..SEQ.$PRO$PFI..PP.D -
NEWNAME         ($PXU..SEQ.$PRO$PFI..OLDPP.D)
ALTER           $PXU..SEQ.$PRO$PFI..NEWPP -
NEWNAME         ($PXU..SEQ.$PRO$PFI..PP)
ALTER           $PXU..SEQ.$PRO$PFI..NEWPP.D -
NEWNAME         ($PXU..SEQ.$PRO$PFI..PP.D)
/+
CA RG$PRO.$PFI.PQ.$SLIT R=Y
/* --- ROTATION FICHER SAUVEGARDE PQ --- */
DEL              ($PXU..SEQ.$PRO$PFI..OLDPQ) CL
ALTER           $PXU..SEQ.$PRO$PFI..PQ -
NEWNAME         ($PXU..SEQ.$PRO$PFI..OLDPQ)
ALTER           $PXU..SEQ.$PRO$PFI..PQ.D -
NEWNAME         ($PXU..SEQ.$PRO$PFI..OLDPQ.D)
ALTER           $PXU..SEQ.$PRO$PFI..NEWPQ -
NEWNAME         ($PXU..SEQ.$PRO$PFI..PQ)
ALTER           $PXU..SEQ.$PRO$PFI..NEWPQ.D -
NEWNAME         ($PXU..SEQ.$PRO$PFI..PQ.D)
/+
/*
// EXEC LISTLOG
/&
* $X EOJ
```

4.16. 3.2 PROCEDURES VSE A CATALOGUER

CATALOGAGE DES PROCEDURES VSE

Le module de JCL '1VSE' contient les sources de procédures de JCL à cataloguer. Ces procédures sont fournies en standard et à titre d'exemple, et rien ne vous empêche de les remplacer par vos propres procédures (par exemple en jouant sur le paramètre 'SORTWK' du job JOB2 de l'implantation).

Dans le flot des procédures à cataloguer se trouvent les procédures VSE suivantes :

SORTWK	Procédure d'allocation d'une SORTWK en SAM sous VSAM
\$0SORTWK	Procédure d'allocation d'une SORTWK en SAM pour la partition BG
\$xSORTWK	Procédure d'allocation d'une SORTWK en SAM pour la partition Fx (F2 à F9)
\$0PACWK1 \$0PACWK2	Procédures d'allocation de fichiers SAM de travail pour la partition BG
\$xPACWK1 \$xPACWK2	Procédures d'allocation de fichiers SAM de travail pour la partition Fx (F2 à F9)

I N S T A L L A T I O N
3.2 PROCEDURES VSE A CATALOGUER

4
16

```

===MOD 1VSE
* $X JOB JNM=$PRFJ.1VSE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.1VSE          ***** $PROD $PREL *****
/* CATALOGAGE DES PROCEDURES VSE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
* ----- STEP 1 ----- LIBR
// EXEC LIBR,PARM=' AC S=$LPRLB$SPRLB'
  CATALP SORTWK
// DLBL SORTWK1,'$XW..SORTWK',0,VSAM,DISP=(NEW,DELETE,DELETE),  $Y.*
  RECSIZE=9999,RECORDS=(999,999)

/+
  CATALP $0SORTWK
// DLBL SORTWK1,'PACWK1.BG',0,DA
// EXTENT SYS001,$VWORK,,1,$WKBG1
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $2SORTWK
// DLBL SORTWK1,'PACWK1.F2',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF21
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $3SORTWK
// DLBL SORTWK1,'PACWK1.F3',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF31
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $4SORTWK
// DLBL SORTWK1,'PACWK1.F4',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF41
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $5SORTWK
// DLBL SORTWK1,'PACWK1.F5',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF51
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $6SORTWK
// DLBL SORTWK1,'PACWK1.F6',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF61
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $7SORTWK
// DLBL SORTWK1,'PACWK1.F7',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF71
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $8SORTWK
// DLBL SORTWK1,'PACWK1.F8',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF81
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $9SORTWK
// DLBL SORTWK1,'PACWK1.F9',0,DA
// EXTENT SYS001,$VWORK,,1,$WKF91
// ASSGN SYS001,DISK,VOL=$VWORK,SHR
/+
  CATALP $0PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK2.BG',0,SD
// EXTENT &ASSGN,$VWORK,,,$WKBG2
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+

```

INSTALLATION
3.2 PROCEDURES VSE A CATALOGUER

4
16

```

CATALP $0PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARAM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARAM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARAM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL, 'PACWK3.BG',0,SD
// EXTENT &ASSGN,$VWORK,,,$WKBG3
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $2PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARAM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARAM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARAM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL, 'PACWK2.F2',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK22
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $2PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARAM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARAM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARAM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL, 'PACWK3.F2',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK23
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $3PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARAM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARAM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARAM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL, 'PACWK2.F3',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK32
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $3PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARAM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARAM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARAM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL, 'PACWK3.F3',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK33
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $4PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARAM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARAM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARAM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL, 'PACWK2.F4',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK42
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $4PACWK2

```

I N S T A L L A T I O N
3.2 PROCEDURES VSE A CATALOGUER

4
 16

```
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK3.F4',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK43
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $5PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK2.F5',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK52
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $5PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK3.F5',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK53
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $6PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK2.F6',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK62
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $6PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK3.F6',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK63
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $7PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK2.F7',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK72
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $7PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
```

INSTALLATION
3.2 PROCEDURES VSE A CATALOGUER

4
16

```
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK3.F7',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK73
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $8PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK2.F8',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK82
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $8PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK3.F8',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK83
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $9PACWK1
// PROC DLBL=SORTWK2,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK2.F9',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK92
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
CATALP $9PACWK2
// PROC DLBL=SORTWK3,ASSGN=SYS001,OPT=TEMP
// IF ASSGN=SYSIPT THEN
// SETPARM OPT=PERM,DLBL=IJSYSIN
// IF ASSGN=SYSPCH THEN
// SETPARM OPT=PERM,DLBL=IJSYSPH
// IF ASSGN=SYSLST THEN
// SETPARM OPT=PERM,DLBL=IJSYSLS
// DLBL &DLBL,'PACWK3.F9',0,SD
// EXTENT &ASSGN,$VWORK,,,$WK93
// ASSGN &ASSGN,DISK,&OPT,VOL=$VWORK,SHR
/+
/*
// EXEC LISTLOG
/&
* $X EOJ
```

4.17. 4.1 DEFINITION DES FICHIERS PERMANENTS

DEFINITION DES FICHIERS PERMANENTS

Le module de JCL '2FIC' effectue un DELETE-DEFINE de l'ensemble des fichiers permanents VSAM après avoir vérifié la non-existence de chaque fichier sous le catalogue indiqué.

I N S T A L L A T I O N

4

4.1 DEFINITION DES FICHIERS PERMANENTS

17

```

===MOD 2FIC
* $X JOB JNM=$PRFJ.2FIC,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.2FIC          ***** $PROD $PREL *****
*
*          DEFINITION DES FICHIERS PERMANENTS
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
===SEQ FOR PEI
LISTC ENT($PXU..$PRO.$PFI.AB)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PFI.AB.SSLIT
END
LISTC ENT($PXU..$PRO.$PFI.AC)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PFI.AC.SSLIT
END
===SEQ
LISTC ENT($PXU..$PRO.$PRO.AE)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PRO.AE.SSLIT
END
LISTC ENT($PXU..$PRO.$PFI.AG)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PFI.AG.SSLIT
END
LISTC ENT($PXU..$PRO.$PFI.AJ)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PFI.AJ.SSLIT
END
LISTC ENT($PXU..$PRO.$PFI.AN)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PFI.AN.SSLIT
END
LISTC ENT($PXU..$PRO.$PRO.AP)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PRO.AP.SSLIT
END
LISTC ENT($PXU..$PRO.$PFI.AR)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PFI.AR.SSLIT
END
LISTC ENT($MODEL F)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DFMODFBK.SSLIT
END
LISTC ENT($MODEL V)
IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DFMODVBK.SSLIT
END
LISTC ENTRY($PXU..SEQ.$PRO$PFI..OLDGL)
IF LASTCC EQ 4 THEN DO
/* ----- DELETE-DEFINE FICHIER GL (-1) ---- */
DEL          ($PXU..SEQ.$PRO$PFI..OLDGL) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..OLDGL) -
            MODEL($MODEL F) -
            RECFM(FB(90)) -
            RECSZ(90 90) REC(1000 1000) -
            VOL($VOLPUD) -
            ) DATA (NAME ($PXU..SEQ.$PRO$PFI..OLDGL.D) )
END
LISTC ENTRY($PXU..SEQ.$PRO$PFI..GL)
IF LASTCC EQ 4 THEN DO
/* ----- DELETE-DEFINE FICHIER GL (0) ---- */
DEL          ($PXU..SEQ.$PRO$PFI..GL) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..GL) -
            MODEL($MODEL F) -

```

I N S T A L L A T I O N

4

4.1 DEFINITION DES FICHIERS PERMANENTS

17

```

                RECFM(FB(90)) -
                RECSZ(90 90) REC(1000 1000) -
                VOL($VOLPUD) -
        ) DATA      (NAME ($PXU..SEQ.$PRO$PFI..GL.D) )
    END
    LISTC ENTRY($PXU..SEQ.$PRO$PFI..OLDGK)
    IF LASTCC EQ 4 THEN DO
        /* ----- DELETE-DEFINE FICHER GK (-1) ----- */
    DEL          ($PXU..SEQ.$PRO$PFI..OLDGK) CL
    DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..OLDGK) -
                MODEL($MODEL) -
                RECFM(FB(100)) -
                RECSZ(100 100) REC(1000 1000) -
                VOL($VOLPUD) -
        ) DATA      (NAME ($PXU..SEQ.$PRO$PFI..OLDGK.D) )
    END
    LISTC ENTRY($PXU..SEQ.$PRO$PFI..GK)
    IF LASTCC EQ 4 THEN DO
        /* ----- DELETE-DEFINE FICHER GK (0) ----- */
    DEL          ($PXU..SEQ.$PRO$PFI..GK) CL
    DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..GK) -
                MODEL($MODEL) -
                RECFM(FB(100)) -
                RECSZ(100 100) REC(1000 1000) -
                VOL($VOLPUD) -
        ) DATA      (NAME ($PXU..SEQ.$PRO$PFI..GK.D) )
    END
    LISTC ENTRY($PXU..$PRO.$PRO.GS)
    IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PRO.GS.$SLIT
    END
    LISTC ENTRY($PXU..$PRO.$PRO.PA)
    IF LASTCC EQ 4 THEN DO
* $X SLI MEM=DF$PRO.$PRO.PA.$SLIT
    END
/*
// EXEC LISTLOG
/&
* $X EOJ

```

4.18. 4.2 PROCEDURE GPRT A CATALOGUER

CATALOGAGE DU JCL DE GPRT

Le module de JCL '2GPR' effectue le catalogage en tant que membre SLI, du JCL d'exécution de la procédure d'édition-génération (GPRT). Cette méthode n'est appliquée en standard qu'aux procédures complexes et d'utilisation fréquente, mais vous pouvez tout-à-fait l'étendre à toutes les procédures VA PAC.

I N S T A L L A T I O N

4

4.2 PROCEDURE GPRT A CATALOGUER

18

```

===MOD 2GPR
* $X JOB JNM=$PRFJ.2GPR,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.2GPR ***** $PROD $PREL *****
/* *****
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// DLBL $LSLI,'$SSLIL',,VSAM
// EXEC LIBR,PARM='AC S=$LSLI$SSLIS;CA GPRT$PRO$PFI..$SLIT R=Y'
/* *****
* GENERATION ET EDITION
/* *****
* $P SLI MEM=DBJB$PRO.$PFI..$SLIT
. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..WGPRT.INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
* $X DATA GPRTDATA
/*
. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL OUT,'$XW..WGPRT.SYSPAF',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO OFILE (OUT) IFILE (SYSIPT) RUS
999999999999
/*
. JCLST04
* --- STEP 04 --- PACB
// DLBL IJSYSUC,'$VCAP',,VSAM
===SEQ FOR PEI
// DLBL PAC7AB,'$PXU..$PRO.$PFI.AB',,VSAM
// DLBL PAC7AC,'$PXU..$PRO.$PFI.AC',,VSAM
===SEQ
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AG,'$PXU..$PRO.$PFI.AG',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AP,'$PXU..$PRO.$PRO.AP',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7EB,'$XW..WGPRT.EB',,VSAM
// DLBL PAC7EE,'$XW..WGPRT.EE',,VSAM
// DLBL PAC7EG,'$XW..WGPRT.EG',,VSAM
// DLBL PAC7EN,'$XW..WGPRT.EN',,VSAM
// DLBL PAC7EP,'$XW..WGPRT.EP',,VSAM
// DLBL PAC7EQ,'$XW..WGPRT.EQ',,VSAM
// DLBL PAC7ER,'$XW..WGPRT.ER',,VSAM
// DLBL PAC7EV,'$XW..WGPRT.EV',,VSAM
// DLBL PAC7GI,'$XW..WGPRT.GI',,VSAM
// DLBL PAC7GK,'$PXU..SEQ.$PRO$PFI..GK',,VSAM
// DLBL PAC7GL,'$PXU..SEQ.$PRO$PFI..GL',,VSAM
// DLBL PAC7GM,'$XW..WGPRT.GM',,VSAM
// DLBL PAC7GN,'$XW..WGPRT.GN',,VSAM
// DLBL PAC7GO,'$XW..WGPRT.GO',,VSAM
// DLBL PAC7GS,'$PXU..$PRO.$PRO.GS',,VSAM
// DLBL PAC7GT,'$XW..WGPRT.GT',,VSAM
// DLBL PAC7G6,'$XW..WGPRT.G6',,VSAM
// DLBL PAC7JG,'$XW..WGPRT.JG',,VSAM
// DLBL PAC7KB,'$XW..WGPRT.KB',,VSAM
// DLBL PAC7KD,'$XW..WGPRT.KD',,VSAM
// DLBL PAC7KE,'$XW..WGPRT.KE',,VSAM
// DLBL PAC7KF,'$XW..WGPRT.KF',,VSAM
// DLBL PAC7KG,'$XW..WGPRT.KG',,VSAM
// DLBL PAC7KM,'$XW..WGPRT.KM',,VSAM
// DLBL PAC7KN,'$XW..WGPRT.KN',,VSAM
// DLBL PAC7KP,'$XW..WGPRT.KP',,VSAM
// DLBL PAC7KQ,'$XW..WGPRT.KQ',,VSAM
// DLBL PAC7KR,'$XW..WGPRT.KR',,VSAM
// DLBL PAC7KS,'$XW..WGPRT.KS',,VSAM
// DLBL PAC7KU,'$XW..WGPRT.KU',,VSAM
// DLBL PAC7KV,'$XW..WGPRT.KV',,VSAM
// DLBL PAC7LG,'$PXU..SEQ.$PRO$PFI..OLDGL',,VSAM
// DLBL PAC7LI,'$XW..WGPRT.LI',,VSAM

```

I N S T A L L A T I O N

4

4.2 PROCEDURE GPRT A CATALOGUER

18

```

// DLBL PAC7LK, '$PXU..SEQ.$PRO$PFI..OLDGK', , VSAM
// DLBL PAC7ME, '$XW..WGPRT.INPUT', , VSAM
// DLBL PAC7MG, '$XW..WGPRT.MG', , VSAM
// DLBL PAC7SC, '$PXY..$PRO.$PRO.SC', , VSAM
// DLBL PAC7SG, '$PXY..$PRO.$PRO.SG', , VSAM
// DLBL PAC7SO, '$XW..WGPRT.SO', , VSAM
// DLBL PAC7SR, '$PXY..$PRO.$PRO.SR', , VSAM
// DLBL PAC7SS, '$PXY..$PRO.$PRO.SS', , VSAM
// DLBL PAC7W1, '$XW..WGPRT.W1', , VSAM
// DLBL PAC7W2, '$XW..WGPRT.W2', , VSAM
// DLBL PAC7W3, '$XW..WGPRT.W3', , VSAM
// DLBL PAC7W4, '$XW..WGPRT.W4', , VSAM
// DLBL PAC7W5, '$XW..WGPRT.W5', , VSAM
// DLBL PAC7W6, '$XW..WGPRT.W6', , VSAM
// DLBL PAC7W7, '$XW..WGPRT.W7', , VSAM
// DLBL PAC7W8, '$XW..WGPRT.W8', , VSAM
// DLBL PAC7W9, '$XW..WGPRT.W9', , VSAM
// DLBL SY8PAF, '$XW..WGPRT.SYSPAF', , VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PACB,SIZE=(PACE80,250K)
/. JCLST05
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC, '$VCAP', , VSAM
// DLBL PAC7EB, '$XW..WGPRT.EB', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7EE, '$XW..WGPRT.EE', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7EG, '$XW..WGPRT.EG', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7EN, '$XW..WGPRT.EN', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7EP, '$XW..WGPRT.EP', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7EQ, '$XW..WGPRT.EQ', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7ER, '$XW..WGPRT.ER', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7EV, '$XW..WGPRT.EV', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7GI, '$XW..WGPRT.GI', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7GM, '$XW..WGPRT.GM', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7GN, '$XW..WGPRT.GN', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7JG, '$XW..WGPRT.JG', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KB, '$XW..WGPRT.KB', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KD, '$XW..WGPRT.KD', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KE, '$XW..WGPRT.KE', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KF, '$XW..WGPRT.KF', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KG, '$XW..WGPRT.KG', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KM, '$XW..WGPRT.KM', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KN, '$XW..WGPRT.KN', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KP, '$XW..WGPRT.KP', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KQ, '$XW..WGPRT.KQ', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KR, '$XW..WGPRT.KR', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KS, '$XW..WGPRT.KS', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KU, '$XW..WGPRT.KU', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7KV, '$XW..WGPRT.KV', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7LI, '$XW..WGPRT.LI', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7ME, '$XW..WGPRT.INPUT', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7MG, '$XW..WGPRT.MG', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W1, '$XW..WGPRT.W1', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W2, '$XW..WGPRT.W2', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W3, '$XW..WGPRT.W3', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W4, '$XW..WGPRT.W4', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W6, '$XW..WGPRT.W6', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W7, '$XW..WGPRT.W7', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W8, '$XW..WGPRT.W8', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL PAC7W9, '$XW..WGPRT.W9', , VSAM, DISP=( ,DELETE,DELETE)
// DLBL SY8PAF, '$XW..WGPRT.SYSPAF', , VSAM, DISP=( ,DELETE,DELETE)
// EXEC IDCAMS,SIZE=AUTO
VERIFY FILE (PAC7EB)
VERIFY FILE (PAC7EE)
VERIFY FILE (PAC7EG)
VERIFY FILE (PAC7EN)
VERIFY FILE (PAC7EP)
VERIFY FILE (PAC7EQ)
VERIFY FILE (PAC7ER)
VERIFY FILE (PAC7EV)
VERIFY FILE (PAC7GI)
VERIFY FILE (PAC7GM)
VERIFY FILE (PAC7GN)
VERIFY FILE (PAC7JG)

```

I N S T A L L A T I O N
4.2 PROCEDURE GPRT A CATALOGUER

PAGE

122

4
18

```
VERIFY FILE (PAC7KB)
VERIFY FILE (PAC7KD)
VERIFY FILE (PAC7KE)
VERIFY FILE (PAC7KF)
VERIFY FILE (PAC7KG)
VERIFY FILE (PAC7KM)
VERIFY FILE (PAC7KN)
VERIFY FILE (PAC7KP)
VERIFY FILE (PAC7KQ)
VERIFY FILE (PAC7KR)
VERIFY FILE (PAC7KS)
VERIFY FILE (PAC7KU)
VERIFY FILE (PAC7KV)
VERIFY FILE (PAC7ME)
VERIFY FILE (PAC7MG)
VERIFY FILE (PAC7LI)
VERIFY FILE (PAC7W1)
VERIFY FILE (PAC7W2)
VERIFY FILE (PAC7W3)
VERIFY FILE (PAC7W4)
VERIFY FILE (PAC7W6)
VERIFY FILE (PAC7W7)
VERIFY FILE (PAC7W8)
VERIFY FILE (PAC7W9)
VERIFY FILE (SY8PAF)
/*
/. JCLST06
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL IN,'$PXU..SEQ.$PRO$PFI..GL',,VSAM
// DLBL OUT,'$PXU..SEQ.$PRO$PFI..OLDGL',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  REPRO OFILE (OUT) IFILE (IN) RUS
/*
/. JCLST07
* --- STEP 07 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL IN,'$PXU..SEQ.$PRO$PFI..GK',,VSAM
// DLBL OUT,'$PXU..SEQ.$PRO$PFI..OLDGK',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  REPRO OFILE (OUT) IFILE (IN) RUS
/*
/+
/&
* $X EOJ
```

4.19. 4.3 PROCEDURE PACX A CATALOGUER

CATALOGAGE DU JCL DE PACX

Le module de JCL '2PAC' effectue le catalogage en tant que membre SLI, du JCL d'exécution de la procédure d'édition-génération (PACX). Cette méthode n'est appliquée en standard qu'aux procédures complexes et d'utilisation fréquente, mais vous pouvez tout-à-fait l'étendre à toutes les procédures VA PAC.

INSTALLATION
4.3 PROCEDURE PACX A CATALOGUER

4
19

```

===MOD 2PAC
* $X JOB JNM=$PRFJ.2PAC,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.2PAC ***** $PROD $PREL *****
/* *****
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// DLBL $LSLI,'$SSLIL',,VSAM
// EXEC LIBR,PARM='AC S=$LSLI$SLIS;CA PACX$PRO$PFI..$SLIT R=Y'
/* *****
* EXTRACTIONS
/* *****
* $P SLI MEM=DBJB$PRO.$PFI..$SLIT
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..WPACX.INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
* $X DATA PACXDATA
/*
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL OUT,'$XW..WPACX.SYSEXT',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO OFILE (OUT) IFILE (SYSIPT) RUS
999999999999
/*
/. JCLST04
* --- STEP 04 --- PACX
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7BM,'$XW..WPACX.BM',,VSAM
// DLBL PAC7GY,'$PXU..RPACX.GY',,VSAM
// DLBL PAC7MB,'$XW..WPACX.INPUT',,VSAM
// DLBL PAC7MJ,'$XW..WPACX.MJ',,VSAM
// DLBL PAC7MM,'$XW..WPACX.MM',,VSAM
// DLBL PAC7MR,'$PXU..RPACX.MR',,VSAM
// DLBL PAC7MV,'$PXU..RPACX.MV',,VSAM
// DLBL PAC7PJ,'$PXU..SEQ.$PRO$PFI..PJ',,VSAM
// DLBL PAC7RE,'$XW..WPACX.RE',,VSAM
// DLBL PAC7RM,'$XW..WPACX.RM',,VSAM
// DLBL PAC7TD,'$PXU..RPACX.TD',,VSAM
// DLBL PAC7UE,'$PXU..RPACX.UE',,VSAM
// DLBL PAC7WD,'$XW..WPACX.WD',,VSAM
// DLBL SY8EXT,'$XW..WPACX.SYSEXT',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PACX,SIZE=(PACS30,250K)
/. JCLST05
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7BM,'$XW..WPACX.BM',,VSAM,DISP=(,DELETE,DELETE)
// DLBL PAC7MJ,'$XW..WPACX.MJ',,VSAM,DISP=(,DELETE,DELETE)
// DLBL PAC7MM,'$XW..WPACX.MM',,VSAM,DISP=(,DELETE,DELETE)
// DLBL PAC7RE,'$XW..WPACX.RE',,VSAM,DISP=(,DELETE,DELETE)
// DLBL PAC7RM,'$XW..WPACX.RM',,VSAM,DISP=(,DELETE,DELETE)
// DLBL PAC7TE,'$XW..WPACX.TE',,VSAM,DISP=(,DELETE,DELETE)
// DLBL PAC7WD,'$XW..WPACX.WD',,VSAM,DISP=(,DELETE,DELETE)
// DLBL SY8EXT,'$XW..WPACX.SYSEXT',,VSAM,DISP=(,DELETE,DELETE)
// EXEC IDCAMS,SIZE=AUTO
VERIFY FILE (PAC7BM)
VERIFY FILE (PAC7MJ)
VERIFY FILE (PAC7MM)
VERIFY FILE (PAC7RE)
VERIFY FILE (PAC7RM)
VERIFY FILE (PAC7TE)
VERIFY FILE (PAC7WD)
VERIFY FILE (SY8EXT)
/*
/+
/&

```

I N S T A L L A T I O N
4.3 PROCEDURE PACX A CATALOGUER

PAGE

125

4
19

* \$X EOJ

4.20. 4.4 DEFINITION DES FICHIERS DE TRAVAIL DE GPRT

DEFINITION DES FICHIERS DE TRAVAIL DE GPRT

Le module de JCL '2WGP' définit préalablement les fichiers de travail de GPRT qui, étant très nombreux, pénaliseraient lourdement les exécutions fréquentes de cette procédure s'ils devaient être définis à chaque passage, ainsi que le fichier de travail pour PACSQL.

Si vous avez utilisé '%' comme premier caractère de l'index des fichiers temporaires (job JOB2, paramètre ===PRM XW), ce qui vous permet la soumission simultanée de GPRT dans plusieurs partitions, vous devrez exécuter la procédure '2WGP' dans chaque partition où les procédures seront susceptibles de s'exécuter. Ces fichiers étant définis avec l'option 'NOALLOCATION', ils ne prennent pas de place disque lorsqu'ils ne sont pas en cours d'utilisation.

I N S T A L L A T I O N

4

4.4 DEFINITION DES FICHIERS DE TRAVAIL DE GPRT

20

```

===MOD 2WGP
* $X JOB JNM=$PRFJ.2WGP,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.2WGP          ***** $PROD $PREL *****
*          DEFINE DES FICHIERS DE TRAVAIL GPRT
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- DEFINE DES FICHIERS DE TRAVAIL GPRT
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DEFINE FICHIER DE TRAVAIL GPRT EB --- */
DEL          ($XW..WGPRT.EB) CL
DEF CL (NAME ($XW..WGPRT.EB) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
        ) DATA(NAME ($XW..WGPRT.EB.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT EE --- */
DEL          ($XW..WGPRT.EE) CL
DEF CL (NAME ($XW..WGPRT.EE) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
        ) DATA(NAME ($XW..WGPRT.EE.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT EG --- */
DEL          ($XW..WGPRT.EG) CL
DEF CL (NAME ($XW..WGPRT.EG) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
        ) DATA(NAME ($XW..WGPRT.EG.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT EN --- */
DEL          ($XW..WGPRT.EN) CL
DEF CL (NAME ($XW..WGPRT.EN) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
        ) DATA(NAME ($XW..WGPRT.EN.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT EP --- */
DEL          ($XW..WGPRT.EP) CL
DEF CL (NAME ($XW..WGPRT.EP) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
        ) DATA(NAME ($XW..WGPRT.EP.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT EQ --- */
DEL          ($XW..WGPRT.EQ) CL
DEF CL (NAME ($XW..WGPRT.EQ) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
        ) DATA(NAME ($XW..WGPRT.EQ.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT ER --- */
DEL          ($XW..WGPRT.ER) CL
DEF CL (NAME ($XW..WGPRT.ER) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
        ) DATA(NAME ($XW..WGPRT.ER.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT EV --- */
DEL          ($XW..WGPRT.EV) CL
DEF CL (NAME ($XW..WGPRT.EV) -

```

I N S T A L L A T I O N

4

4.4 DEFINITION DES FICHIERS DE TRAVAIL DE GPRT

20

```

        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.EV.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT GI --- */
DEL      ($XW..WGPRT.GI) CL
DEF CL (NAME ($XW..WGPRT.GI) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(080 080) -
        RECFM(FB(080)) -
    ) DATA(NAME ($XW..WGPRT.GI.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT GM --- */
DEL      ($XW..WGPRT.GM) CL
DEF CL (NAME ($XW..WGPRT.GM) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(080 080) -
        RECFM(FB(080)) -
    ) DATA(NAME ($XW..WGPRT.GM.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT GN --- */
DEL      ($XW..WGPRT.GN) CL
DEF CL (NAME ($XW..WGPRT.GN) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(265 265) -
        RECFM(FB(265)) -
    ) DATA(NAME ($XW..WGPRT.GN.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT GO --- */
DEL      ($XW..WGPRT.GO) CL
DEF CL (NAME ($XW..WGPRT.GO) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(204 204) -
        RECFM(FB(204)) -
    ) DATA(NAME ($XW..WGPRT.GO.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT GT --- */
DEL      ($XW..WGPRT.GT) CL
DEF CL (NAME ($XW..WGPRT.GT) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.GT.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT INPUT --- */
DEL      ($XW..WGPRT.INPUT) CL
DEF CL (NAME ($XW..WGPRT.INPUT) -
        MODEL($MODEL) -
        REC (0010 0010) -
        RECSZ(080 080) -
        RECFM(FB(080)) -
    ) DATA(NAME ($XW..WGPRT.INPUT.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT JG --- */
DEL      ($XW..WGPRT.JG) CL
DEF CL (NAME ($XW..WGPRT.JG) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(080 080) -
        RECFM(FB(080)) -
    ) DATA(NAME ($XW..WGPRT.JG.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT KB --- */
DEL      ($XW..WGPRT.KB) CL
DEF CL (NAME ($XW..WGPRT.KB) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KB.D) )
/* --- DEFINE FICHIER DE TRAVAIL GPRT KD --- */
DEL      ($XW..WGPRT.KD) CL
DEF CL (NAME ($XW..WGPRT.KD) -
        MODEL($MODEL) -

```

I N S T A L L A T I O N

4

4.4 DEFINITION DES FICHIERS DE TRAVAIL DE GPRT

20

```

        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KD.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KE --- */
DEL      ($XW..WGPRT.KE) CL
DEF CL (NAME ($XW..WGPRT.KE) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KE.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KF --- */
DEL      ($XW..WGPRT.KF) CL
DEF CL (NAME ($XW..WGPRT.KF) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KF.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KG --- */
DEL      ($XW..WGPRT.KG) CL
DEF CL (NAME ($XW..WGPRT.KG) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KG.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KM --- */
DEL      ($XW..WGPRT.KM) CL
DEF CL (NAME ($XW..WGPRT.KM) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KM.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KN --- */
DEL      ($XW..WGPRT.KN) CL
DEF CL (NAME ($XW..WGPRT.KN) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KN.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KP --- */
DEL      ($XW..WGPRT.KP) CL
DEF CL (NAME ($XW..WGPRT.KP) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KP.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KQ --- */
DEL      ($XW..WGPRT.KQ) CL
DEF CL (NAME ($XW..WGPRT.KQ) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KQ.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KR --- */
DEL      ($XW..WGPRT.KR) CL
DEF CL (NAME ($XW..WGPRT.KR) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KR.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KS --- */
DEL      ($XW..WGPRT.KS) CL
DEF CL (NAME ($XW..WGPRT.KS) -
        MODEL($MODEL) -
        REC (6000 6000) -

```

I N S T A L L A T I O N

4

4.4 DEFINITION DES FICHIERS DE TRAVAIL DE GPRT

20

```

        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KS.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KU --- */
DEL      ($XW..WGPRT.KU) CL
DEF CL (NAME ($XW..WGPRT.KU) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KU.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT KV --- */
DEL      ($XW..WGPRT.KV) CL
DEF CL (NAME ($XW..WGPRT.KV) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.KV.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT LI --- */
DEL      ($XW..WGPRT.LI) CL
DEF CL (NAME ($XW..WGPRT.LI) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(080 080) -
        RECFM(FB(080)) -
    ) DATA(NAME ($XW..WGPRT.LI.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT MG --- */
DEL      ($XW..WGPRT.MG) CL
DEF CL (NAME ($XW..WGPRT.MG) -
        MODEL($MODEL) -
        REC (3000 3000) -
        RECSZ(150 150) -
        RECFM(FB(150)) -
    ) DATA(NAME ($XW..WGPRT.MG.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT SO --- */
DEL      ($XW..WGPRT.SO) CL
DEF CL (NAME ($XW..WGPRT.SO) -
        MODEL($MODEL) -
        REC (6000 6000) -
        RECSZ(312 312) -
        RECFM(FB(312)) -
    ) DATA(NAME ($XW..WGPRT.SO.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT SYSPAF --- */
DEL      ($XW..WGPRT.SYSPAF) CL
DEF CL (NAME ($XW..WGPRT.SYSPAF) -
        SHR(2 3) RUS NAL KEYS(12 0) -
        REC (1000 1000) -
        RECSZ(170 468) -
        VOL($VOLVWK) -
    ) INDEX(NAME ($XW..WGPRT.SYSPAF.I) -
    ) DATA (NAME ($XW..WGPRT.SYSPAF.D) CISZ(8192) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W1 --- */
DEL      ($XW..WGPRT.W1) CL
DEF CL (NAME ($XW..WGPRT.W1) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.W1.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W2 --- */
DEL      ($XW..WGPRT.W2) CL
DEF CL (NAME ($XW..WGPRT.W2) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.W2.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W3 --- */
DEL      ($XW..WGPRT.W3) CL
DEF CL (NAME ($XW..WGPRT.W3) -
        MODEL($MODEL) -
        REC (2000 2000) -

```

I N S T A L L A T I O N

4

4.4 DEFINITION DES FICHIERS DE TRAVAIL DE GPRT

20

```
        RECSZ(180 180) -
        RECFM(FB(180)) -
    ) DATA(NAME ($XW..WGPRT.W3.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W4 --- */
DEL      ($XW..WGPRT.W4) CL
DEF CL (NAME ($XW..WGPRT.W4) -
        MODEL($MODEL) -
        REC (5000 5000) -
        RECSZ(090 090) -
        RECFM(FB(090)) -
    ) DATA(NAME ($XW..WGPRT.W4.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W6 --- */
DEL      ($XW..WGPRT.W6) CL
DEF CL (NAME ($XW..WGPRT.W6) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(308 308) -
        RECFM(FB(308)) -
    ) DATA(NAME ($XW..WGPRT.W6.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W7 --- */
DEL      ($XW..WGPRT.W7) CL
DEF CL (NAME ($XW..WGPRT.W7) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(308 308) -
        RECFM(FB(308)) -
    ) DATA(NAME ($XW..WGPRT.W7.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W8 --- */
DEL      ($XW..WGPRT.W8) CL
DEF CL (NAME ($XW..WGPRT.W8) -
        MODEL($MODEL) -
        REC (2000 2000) -
        RECSZ(308 308) -
        RECFM(FB(308)) -
    ) DATA(NAME ($XW..WGPRT.W8.D) )
/* --- DEFINE FICHER DE TRAVAIL GPRT W9 --- */
DEL      ($XW..WGPRT.W9) CL
DEF CL (NAME ($XW..WGPRT.W9) -
        MODEL($MODEL) -
        REC (9000 9000) -
        RECSZ(055 055) -
        RECFM(FB(055)) -
    ) DATA(NAME ($XW..WGPRT.W9.D) )
/*
// EXEC LISTLOG
/&
* $X EOJ
```

4.21. 4.5 DEFINITION DES FICHIERS DE TRAVAIL DE PACX

DEFINITION DES FICHIERS DE TRAVAIL DE PACX

Le module de JCL '2WPA' définit préalablement les fichiers de travail de PACX qui, étant très nombreux, pénaliseraient lourdement les exécutions fréquentes de cette procédure s'ils devaient être définis à chaque passage.

Si vous avez utilisé '%' comme premier caractère de l'index des fichiers temporaires (job JOB2, paramètre ===PRM XW), ce qui vous permet la soumission simultanée de GPRT dans plusieurs partitions, vous devrez exécuter la procédure '2WPA' dans chaque partition où les procédures seront susceptibles de s'exécuter. Ces fichiers étant définis avec l'option 'NOALLOCATION', ils ne prennent pas de place disque lorsqu'ils ne sont pas en cours d'utilisation.

I N S T A L L A T I O N

4

4.5 DEFINITION DES FICHIERS DE TRAVAIL DE PACX

21

```

===MOD 2WPA
* $X JOB JNM=$PRFJ.2WPA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.2WPA          ***** $PROD $PREL *****
*          DEFINE DES FICHIERS DE TRAVAIL PACX
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- DEFINE DES FICHIERS DE TRAVAIL PACX
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DEFINE FICHIER DE TRAVAIL PACX BM --- */
DEL          ($XW..WPACX.BM) CL
DEF CL (NAME ($XW..WPACX.BM) -
        MODEL($MODEL) -
        REC (0010 0010) -
        RECSZ(080 080) -
        RECFM(FB(080)) -
        ) DATA(NAME ($XW..WPACX.BM.D) )
/* --- DEFINE FICHIER DE TRAVAIL PACX INPUT --- */
DEL          ($XW..WPACX.INPUT) CL
DEF CL (NAME ($XW..WPACX.INPUT) -
        MODEL($MODEL) -
        REC (0010 0010) -
        RECSZ(080 080) -
        RECFM(FB(080)) -
        ) DATA(NAME ($XW..WPACX.INPUT.D) )
/* --- DEFINE FICHIER DE TRAVAIL PACX MJ --- */
DEL          ($XW..WPACX.MJ) CL
DEF CL (NAME ($XW..WPACX.MJ) -
        MODEL($MODEL) -
        REC (1000 1000) -
        RECSZ(152 152) -
        RECFM(FB(152)) -
        ) DATA(NAME ($XW..WPACX.MJ.D) )
/* --- DEFINE FICHIER DE TRAVAIL PACX MM --- */
DEL          ($XW..WPACX.MM) CL
DEF CL (NAME ($XW..WPACX.MM) -
        MODEL($MODEL) -
        REC (1000 1000) -
        RECSZ(055 055) -
        RECFM(FB(055)) -
        ) DATA(NAME ($XW..WPACX.MM.D) )
/* --- DEFINE FICHIER DE TRAVAIL PACX RE --- */
DEL          ($XW..WPACX.RE) CL
DEF CL (NAME ($XW..WPACX.RE) -
        MODEL($MODEL) -
        REC (1000 1000) -
        RECSZ(012 012) -
        RECFM(FB(012)) -
        ) DATA(NAME ($XW..WPACX.RE.D) )
/* --- DEFINE FICHIER DE TRAVAIL PACX RM --- */
DEL          ($XW..WPACX.RM) CL
DEF CL (NAME ($XW..WPACX.RM) -
        MODEL($MODEL) -
        REC (1000 1000) -
        RECSZ(167 167) -
        RECFM(FB(167)) -
        ) DATA(NAME ($XW..WPACX.RM.D) )
/* --- DEFINE FICHIER DE TRAVAIL PACX SYSEXT --- */
DEL          ($XW..WPACX.SYSEXT) CL
DEF CL (NAME ($XW..WPACX.SYSEXT) -
        SHR(2 3) RUS NAL KEYS(12 0) -
        REC (1000 1000) -
        RECSZ(112 112) -
        VOL($VOLVWK) -
        ) INDEX(NAME ($XW..WPACX.SYSEXT.I) -
        ) DATA (NAME ($XW..WPACX.SYSEXT.D) Cisz(8192) )
/* --- DEFINE FICHIER DE TRAVAIL PACX TE --- */
DEL          ($XW..WPACX.TE) CL

```

I N S T A L L A T I O N
4.5 DEFINITION DES FICHIERS DE TRAVAIL DE PACX

PAGE

134

4

21

```
DEF CL (NAME ($XW..WPACX.TE) -  
        MODEL($MODELFB) -  
        REC (1000 1000) -  
        RECSZ(180 180) -  
        RECFM(FB(180)) -  
        ) DATA(NAME ($XW..WPACX.TE.D) )  
/* --- DEFINE FICHIER DE TRAVAIL PACX WD --- */  
DEL ($XW..WPACX.WD) CL  
DEF CL (NAME ($XW..WPACX.WD) -  
        MODEL($MODELFB) -  
        REC (1000 1000) -  
        RECSZ(167 167) -  
        RECFM(FB(167)) -  
        ) DATA(NAME ($XW..WPACX.WD.D) )  
/*  
// EXEC LISTLOG  
/&  
* $X EOJ
```

4.22. 4.6 INSTALLATION DES MODULES EXECUTABLES

INSTALLATION DES MODULES EXECUTABLES

Exécuter le job '2PHA'.

I N S T A L L A T I O N

4

4.6 INSTALLATION DES MODULES EXECUTABLES

22

```

===MOD 2PHA
* $X JOB JNM=$PRFJ.2PHA,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.2PHA          ***** $PROD $PREL *****
*   INSTALLATION DES MODULES (RESTORE)
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// DLBL $LMODO,'$MODO',,VSAM
/* GOTO JCLSTXX
* --- STEP 1 --- LIBR
// ASSGN SYS004,$TAP1
// PAUSE MONTER BANDE $TAPEI SUR $TAP1
// MTC REW,SYS004
// MTC FSF,SYS004,20
// EXEC LIBR
ACCESS SUBLIB=$LMODO$SMODO
RENAME $PRO.*.PHASE:D2*.PHASE
RESTORE -
  CGI.LB.PACA05.PHASE:$LMODB$SMODB -
  CGI.LB.PACA10.PHASE:$LMODB$SMODB -
  CGI.LB.PACA15.PHASE:$LMODB$SMODB -
  CGI.LB.PACA20.PHASE:$LMODB$SMODB -
  CGI.LB.PACB.PHASE:$LMODB$SMODB -
  CGI.LB.PACB30.PHASE:$LMODB$SMODB -
===SEQ FOR SQL
  CGI.LB.PACB31.PHASE:$LMODB$SMODB -
===SEQ
  CGI.LB.PACB40.PHASE:$LMODB$SMODB -
  CGI.LB.PACB80.PHASE:$LMODB$SMODB -
  CGI.LB.PACCTL.PHASE:$LMODB$SMODB -
  CGI.LB.PACC30.PHASE:$LMODB$SMODB -
  CGI.LB.PACC40.PHASE:$LMODB$SMODB -
  CGI.LB.PACC80.PHASE:$LMODB$SMODB -
  CGI.LB.PACDTP.PHASE:$LMODB$SMODB -
  CGI.LB.PACD30.PHASE:$LMODB$SMODB -
  CGI.LB.PACD40.PHASE:$LMODB$SMODB -
  CGI.LB.PACD80.PHASE:$LMODB$SMODB -
  CGI.LB.PACD90.PHASE:$LMODB$SMODB -
  CGI.LB.PACE30.PHASE:$LMODB$SMODB -
  CGI.LB.PACE40.PHASE:$LMODB$SMODB -
  CGI.LB.PACE80.PHASE:$LMODB$SMODB -
  CGI.LB.PACFGY.PHASE:$LMODB$SMODB -
  CGI.LB.PACFMB.PHASE:$LMODB$SMODB -
  CGI.LB.PACFTD.PHASE:$LMODB$SMODB -
  CGI.LB.PACG3C.PHASE:$LMODB$SMODB -
  CGI.LB.PACG3S.PHASE:$LMODB$SMODB -
  CGI.LB.PACG4S.PHASE:$LMODB$SMODB -
  CGI.LB.PACG8C.PHASE:$LMODB$SMODB -
  CGI.LB.PACG8S.PHASE:$LMODB$SMODB -
===SEQ FOR VISP
  CGI.LB.PACINS.PHASE:$LMODB$SMODB -
===SEQ
  CGI.LB.PACK30.PHASE:$LMODB$SMODB -
  CGI.LB.PACK80.PHASE:$LMODB$SMODB -
  CGI.LB.PACK90.PHASE:$LMODB$SMODB -
  CGI.LB.PACL30.PHASE:$LMODB$SMODB -
  CGI.LB.PACL80.PHASE:$LMODB$SMODB -
  CGI.LB.PACL90.PHASE:$LMODB$SMODB -
  CGI.LB.PACL92.PHASE:$LMODB$SMODB -
  CGI.LB.PACL93.PHASE:$LMODB$SMODB -
  CGI.LB.PACL95.PHASE:$LMODB$SMODB -
  CGI.LB.PACM30.PHASE:$LMODB$SMODB -
  CGI.LB.PACM80.PHASE:$LMODB$SMODB -
  CGI.LB.PACN25.PHASE:$LMODB$SMODB -
  CGI.LB.PACN30.PHASE:$LMODB$SMODB -
  CGI.LB.PACN35.PHASE:$LMODB$SMODB -
  CGI.LB.PACN40.PHASE:$LMODB$SMODB -
  CGI.LB.PACN80.PHASE:$LMODB$SMODB -
  CGI.LB.PACP30.PHASE:$LMODB$SMODB -
  CGI.LB.PACP40.PHASE:$LMODB$SMODB -
  CGI.LB.PACP80.PHASE:$LMODB$SMODB -

```

I N S T A L L A T I O N

4

4.6 INSTALLATION DES MODULES EXECUTABLES

22

```

CGI.LB.PACP82.PHASE:$LMOB$$SMOBD -
CGI.LB.PACQ.PHASE:$LMOB$$SMOBD -
===SEQ FOR PEI
CGI.LB.PACR01.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR10.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR20.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR22.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR30.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR40.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR60.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR61.PHASE:$LMOB$$SMOBD -
CGI.LB.PACR90.PHASE:$LMOB$$SMOBD -
===SEQ
CGI.LB.PACSJO.PHASE:$LMOB$$SMOBD -
CGI.LB.PACSMO.PHASE:$LMOB$$SMOBD -
CGI.LB.PACSPU.PHASE:$LMOB$$SMOBD -
CGI.LB.PACSRM.PHASE:$LMOB$$SMOBD -
CGI.LB.PACS30.PHASE:$LMOB$$SMOBD -
CGI.LB.PACS40.PHASE:$LMOB$$SMOBD -
CGI.LB.PACS50.PHASE:$LMOB$$SMOBD -
CGI.LB.PACS60.PHASE:$LMOB$$SMOBD -
CGI.LB.PACS75.PHASE:$LMOB$$SMOBD -
CGI.LB.PACS80.PHASE:$LMOB$$SMOBD -
===SEQ FOR TAB
CGI.LB.PACTIN.PHASE:$LMOB$$SMOBD -
CGI.LB.PACTI1.PHASE:$LMOB$$SMOBD -
CGI.LB.PACT40.PHASE:$LMOB$$SMOBD -
CGI.LB.PACT41.PHASE:$LMOB$$SMOBD -
CGI.LB.PACT45.PHASE:$LMOB$$SMOBD -
CGI.LB.PACT50.PHASE:$LMOB$$SMOBD -
CGI.LB.PACT51.PHASE:$LMOB$$SMOBD -
===SEQ
CGI.LB.PACU15.PHASE:$LMOB$$SMOBD -
CGI.LB.PACU80.PHASE:$LMOB$$SMOBD -
CGI.LB.PACU99.PHASE:$LMOB$$SMOBD -
CGI.LB.PACX.PHASE:$LMOB$$SMOBD -
CGI.LB.PADM10.PHASE:$LMOB$$SMOBD -
CGI.LB.PAFP10.PHASE:$LMOB$$SMOBD -
CGI.LB.PAF900.PHASE:$LMOB$$SMOBD -
===SEQ FOR S2K
CGI.LB.PAN200.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN205.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN210.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN212.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN215.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN220.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN230.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN240.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN250.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN255.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN260.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN270.PHASE:$LMOB$$SMOBD -
CGI.LB.PAN280.PHASE:$LMOB$$SMOBD -
===SEQ
CGI.LB.PBBTST.PHASE:$LMOB$$SMOBD -
CGI.LB.PBBTWS.PHASE:$LMOB$$SMOBD -
CGI.LB.PBBT98.PHASE:$LMOB$$SMOBD -
===SEQ FOR PACTR
CGI.LB.PDS600.PHASE:$LMOB$$SMOBD -
CGI.LB.PDS610.PHASE:$LMOB$$SMOBD -
===SEQ
CGI.LB.PREI00.PHASE:$LMOB$$SMOBD -
CGI.LB.PREI40.PHASE:$LMOB$$SMOBD -
CGI.LB.PREI50.PHASE:$LMOB$$SMOBD -
CGI.LB.PRE986.PHASE:$LMOB$$SMOBD -
===SEQ FOR TAB
CGI.LB.PTARSD.PHASE:$LMOB$$SMOBD -
CGI.LB.PTAR20.PHASE:$LMOB$$SMOBD -
CGI.LB.PTASVD.PHASE:$LMOB$$SMOBD -
===SEQ
===SEQ FOR TEAM
CGI.LB.PTC010.PHASE:$LMOB$$SMOBD -
CGI.LB.PTC030.PHASE:$LMOB$$SMOBD -

```

I N S T A L L A T I O N

4

4.6 INSTALLATION DES MODULES EXECUTABLES

22

```
CGI.LB.PTC100.PHASE:$LMOBBSMOB -
CGI.LB.PTC200.PHASE:$LMOBBSMOB -
CGI.LB.PTC220.PHASE:$LMOBBSMOB -
CGI.LB.PTC400.PHASE:$LMOBBSMOB -
CGI.LB.PTC440.PHASE:$LMOBBSMOB -
===SEQ
CGI.LB.PTED30.PHASE:$LMOBBSMOB -
CGI.LB.PTED60.PHASE:$LMOBBSMOB -
CGI.LB.PTEP90.PHASE:$LMOBBSMOB -
CGI.LB.PTEXD0.PHASE:$LMOBBSMOB -
CGI.LB.PTEX30.PHASE:$LMOBBSMOB -
CGI.LB.PTEX80.PHASE:$LMOBBSMOB -
CGI.LB.PTUBAS.PHASE:$LMOBBSMOB -
CGI.LB.PTUCR1.PHASE:$LMOBBSMOB -
CGI.LB.PTUCR2.PHASE:$LMOBBSMOB -
CGI.LB.PTUCSS.PHASE:$LMOBBSMOB -
CGI.LB.PTUD10.PHASE:$LMOBBSMOB -
CGI.LB.PTUD20.PHASE:$LMOBBSMOB -
CGI.LB.PTUD30.PHASE:$LMOBBSMOB -
CGI.LB.PTUESS.PHASE:$LMOBBSMOB -
===SEQ FOR PACTR
CGI.LB.PTUG05.PHASE:$LMOBBSMOB -
CGI.LB.PTUG06.PHASE:$LMOBBSMOB -
CGI.LB.PTUG07.PHASE:$LMOBBSMOB -
CGI.LB.PTUG10.PHASE:$LMOBBSMOB -
CGI.LB.PTUG11.PHASE:$LMOBBSMOB -
CGI.LB.PTUG12.PHASE:$LMOBBSMOB -
CGI.LB.PTUG42.PHASE:$LMOBBSMOB -
CGI.LB.PTUG44.PHASE:$LMOBBSMOB -
CGI.LB.PTUG46.PHASE:$LMOBBSMOB -
CGI.LB.PTUG50.PHASE:$LMOBBSMOB -
CGI.LB.PTUG60.PHASE:$LMOBBSMOB -
CGI.LB.PTUG61.PHASE:$LMOBBSMOB -
CGI.LB.PTUG90.PHASE:$LMOBBSMOB -
===SEQ
CGI.LB.PTULOI.PHASE:$LMOBBSMOB -
CGI.LB.PTULVB.PHASE:$LMOBBSMOB -
CGI.LB.PTUN00.PHASE:$LMOBBSMOB -
CGI.LB.PTUN10.PHASE:$LMOBBSMOB -
CGI.LB.PTUN40.PHASE:$LMOBBSMOB -
CGI.LB.PTUQ10.PHASE:$LMOBBSMOB -
CGI.LB.PTUQ15.PHASE:$LMOBBSMOB -
CGI.LB.PTUQ20.PHASE:$LMOBBSMOB -
CGI.LB.PTUQ24.PHASE:$LMOBBSMOB -
CGI.LB.PTUQ25.PHASE:$LMOBBSMOB -
CGI.LB.PTUQ30.PHASE:$LMOBBSMOB -
CGI.LB.PTUR00.PHASE:$LMOBBSMOB -
CGI.LB.PTU001.PHASE:$LMOBBSMOB -
CGI.LB.PTU004.PHASE:$LMOBBSMOB -
CGI.LB.PTU100.PHASE:$LMOBBSMOB -
CGI.LB.PTU120.PHASE:$LMOBBSMOB -
CGI.LB.PTU130.PHASE:$LMOBBSMOB -
CGI.LB.PTU140.PHASE:$LMOBBSMOB -
CGI.LB.PTU2CL.PHASE:$LMOBBSMOB -
CGI.LB.PTU200.PHASE:$LMOBBSMOB -
CGI.LB.PTU208.PHASE:$LMOBBSMOB -
CGI.LB.PTU210.PHASE:$LMOBBSMOB -
CGI.LB.PTU220.PHASE:$LMOBBSMOB -
CGI.LB.PTU240.PHASE:$LMOBBSMOB -
CGI.LB.PTU300.PHASE:$LMOBBSMOB -
CGI.LB.PTU320.PHASE:$LMOBBSMOB -
CGI.LB.PTU380.PHASE:$LMOBBSMOB -
CGI.LB.PTU400.PHASE:$LMOBBSMOB -
CGI.LB.PTU402.PHASE:$LMOBBSMOB -
CGI.LB.PTU420.PHASE:$LMOBBSMOB -
CGI.LB.PTU500.PHASE:$LMOBBSMOB -
CGI.LB.PTU502.PHASE:$LMOBBSMOB -
CGI.LB.PTU550.PHASE:$LMOBBSMOB -
CGI.LB.PTU560.PHASE:$LMOBBSMOB -
CGI.LB.PTU630.PHASE:$LMOBBSMOB -
CGI.LB.PTU640.PHASE:$LMOBBSMOB -
CGI.LB.PTU810.PHASE:$LMOBBSMOB -
CGI.LB.PTU815.PHASE:$LMOBBSMOB -
```

I N S T A L L A T I O N

4

4.6 INSTALLATION DES MODULES EXECUTABLES

22

```
CGI.LB.PTU850.PHASE:$LMOB$SMOBD -
CGI.LB.PTU855.PHASE:$LMOB$SMOBD -
===SEQ FOR ROLD
CGI.LB.PTU908.PHASE:$LMOB$SMOBD -
===SEQ
===SEQ FOR VISF
CGI.LB.PVA100.PHASE:$LMOB$SMOBD -
CGI.LB.PVA110.PHASE:$LMOB$SMOBD -
CGI.LB.PVA300.PHASE:$LMOB$SMOBD -
CGI.LB.PVA305.PHASE:$LMOB$SMOBD -
CGI.LB.PVA310.PHASE:$LMOB$SMOBD -
CGI.LB.PVA320.PHASE:$LMOB$SMOBD -
CGI.LB.PVA400.PHASE:$LMOB$SMOBD -
===SEQ
CGI.LB.PYSMCC.PHASE:$LMOB$SMOBD -
CGI.LB.PYSMC2.PHASE:$LMOB$SMOBD -
CGI.LB.PYSMC3.PHASE:$LMOB$SMOBD -
===SEQ FOR ROLD
CGI.LB.REP2PJ.PHASE:$LMOB$SMOBD -
===SEQ
CGI.LB.UTIXSR.PHASE:$LMOB$SMOBD -
===SEQ FOR PUF
CGI.LB.D2FANM.PHASE:$LMODO$SMODO -
CGI.LB.D2FBIB.PHASE:$LMODO$SMODO -
CGI.LB.D2FCTL.PHASE:$LMODO$SMODO -
CGI.LB.D2FDBD.PHASE:$LMODO$SMODO -
CGI.LB.D2FECD.PHASE:$LMODO$SMODO -
CGI.LB.D2FENU.PHASE:$LMODO$SMODO -
CGI.LB.D2FFOG.PHASE:$LMODO$SMODO -
CGI.LB.D2FMCL.PHASE:$LMODO$SMODO -
CGI.LB.D2FPGM.PHASE:$LMODO$SMODO -
CGI.LB.D2FRUB.PHASE:$LMODO$SMODO -
CGI.LB.D2FSDO.PHASE:$LMODO$SMODO -
CGI.LB.D2FTXT.PHASE:$LMODO$SMODO -
CGI.LB.D2FVER.PHASE:$LMODO$SMODO -
CGI.LB.D2F000.PHASE:$LMODO$SMODO -
===SEQ
CGI.LB.D2PA00.PHASE:$LMODO$SMODO -
CGI.LB.D2PA01.PHASE:$LMODO$SMODO -
CGI.LB.D2PA10.PHASE:$LMODO$SMODO -
CGI.LB.D2PA11.PHASE:$LMODO$SMODO -
CGI.LB.D2PA12.PHASE:$LMODO$SMODO -
CGI.LB.D2PA13.PHASE:$LMODO$SMODO -
CGI.LB.D2PA14.PHASE:$LMODO$SMODO -
CGI.LB.D2PA15.PHASE:$LMODO$SMODO -
CGI.LB.D2PA16.PHASE:$LMODO$SMODO -
CGI.LB.D2PA17.PHASE:$LMODO$SMODO -
CGI.LB.D2PA18.PHASE:$LMODO$SMODO -
CGI.LB.D2PA19.PHASE:$LMODO$SMODO -
CGI.LB.D2PA20.PHASE:$LMODO$SMODO -
CGI.LB.D2PA21.PHASE:$LMODO$SMODO -
CGI.LB.D2PA22.PHASE:$LMODO$SMODO -
===SEQ FOR PEI
CGI.LB.D2PA30.PHASE:$LMODO$SMODO -
CGI.LB.D2PA31.PHASE:$LMODO$SMODO -
CGI.LB.D2PA32.PHASE:$LMODO$SMODO -
CGI.LB.D2PA33.PHASE:$LMODO$SMODO -
CGI.LB.D2PA34.PHASE:$LMODO$SMODO -
CGI.LB.D2PA35.PHASE:$LMODO$SMODO -
===SEQ
CGI.LB.D2PBND.PHASE:$LMODO$SMODO -
CGI.LB.D2PPHP.PHASE:$LMODO$SMODO -
CGI.LB.D2QA00.PHASE:$LMODO$SMODO -
CGI.LB.D2QB00.PHASE:$LMODO$SMODO -
CGI.LB.D2QC00.PHASE:$LMODO$SMODO -
CGI.LB.D2QC01.PHASE:$LMODO$SMODO -
CGI.LB.D2QC50.PHASE:$LMODO$SMODO -
CGI.LB.D2QD00.PHASE:$LMODO$SMODO -
CGI.LB.D2QE00.PHASE:$LMODO$SMODO -
CGI.LB.D2QF00.PHASE:$LMODO$SMODO -
CGI.LB.D2QF10.PHASE:$LMODO$SMODO -
CGI.LB.D2QG00.PHASE:$LMODO$SMODO -
CGI.LB.D2QH00.PHASE:$LMODO$SMODO -
```

I N S T A L L A T I O N

4

4.6 INSTALLATION DES MODULES EXECUTABLES

22

```
CGI.LB.D2QH01.PHASE:$LMODO$SMODO -
CGI.LB.D2QH20.PHASE:$LMODO$SMODO -
CGI.LB.D2QH30.PHASE:$LMODO$SMODO -
CGI.LB.D2QI00.PHASE:$LMODO$SMODO -
CGI.LB.D2QI01.PHASE:$LMODO$SMODO -
CGI.LB.D2QI02.PHASE:$LMODO$SMODO -
CGI.LB.D2QI03.PHASE:$LMODO$SMODO -
CGI.LB.D2QI04.PHASE:$LMODO$SMODO -
CGI.LB.D2QI05.PHASE:$LMODO$SMODO -
CGI.LB.D2QI20.PHASE:$LMODO$SMODO -
CGI.LB.D2QI21.PHASE:$LMODO$SMODO -
CGI.LB.D2QI50.PHASE:$LMODO$SMODO -
CGI.LB.D2QK10.PHASE:$LMODO$SMODO -
CGI.LB.D2QK20.PHASE:$LMODO$SMODO -
CGI.LB.D2QK30.PHASE:$LMODO$SMODO -
CGI.LB.D2QL10.PHASE:$LMODO$SMODO -
CGI.LB.D2QL20.PHASE:$LMODO$SMODO -
CGI.LB.D2QL21.PHASE:$LMODO$SMODO -
CGI.LB.D2QL30.PHASE:$LMODO$SMODO -
===SEQ FOR SQL
CGI.LB.D2QL40.PHASE:$LMODO$SMODO -
CGI.LB.D2QL41.PHASE:$LMODO$SMODO -
CGI.LB.D2QL45.PHASE:$LMODO$SMODO -
CGI.LB.D2QL46.PHASE:$LMODO$SMODO -
===SEQ
CGI.LB.D2QM00.PHASE:$LMODO$SMODO -
CGI.LB.D2QP00.PHASE:$LMODO$SMODO -
CGI.LB.D2QP01.PHASE:$LMODO$SMODO -
CGI.LB.D2QP02.PHASE:$LMODO$SMODO -
CGI.LB.D2QP03.PHASE:$LMODO$SMODO -
CGI.LB.D2QP04.PHASE:$LMODO$SMODO -
CGI.LB.D2QP05.PHASE:$LMODO$SMODO -
CGI.LB.D2QP06.PHASE:$LMODO$SMODO -
CGI.LB.D2QP07.PHASE:$LMODO$SMODO -
CGI.LB.D2QP08.PHASE:$LMODO$SMODO -
CGI.LB.D2QP50.PHASE:$LMODO$SMODO -
CGI.LB.D2QR00.PHASE:$LMODO$SMODO -
CGI.LB.D2QS02.PHASE:$LMODO$SMODO -
CGI.LB.D2QS03.PHASE:$LMODO$SMODO -
CGI.LB.D2QS04.PHASE:$LMODO$SMODO -
CGI.LB.D2QS05.PHASE:$LMODO$SMODO -
CGI.LB.D2QS06.PHASE:$LMODO$SMODO -
CGI.LB.D2QS08.PHASE:$LMODO$SMODO -
CGI.LB.D2QS09.PHASE:$LMODO$SMODO -
CGI.LB.D2QS10.PHASE:$LMODO$SMODO -
CGI.LB.D2QT00.PHASE:$LMODO$SMODO -
CGI.LB.D2QT10.PHASE:$LMODO$SMODO -
CGI.LB.D2QT20.PHASE:$LMODO$SMODO -
CGI.LB.D2QT50.PHASE:$LMODO$SMODO -
CGI.LB.D2QUPA.PHASE:$LMODO$SMODO -
CGI.LB.D2QUPF.PHASE:$LMODO$SMODO -
CGI.LB.D2QU00.PHASE:$LMODO$SMODO -
CGI.LB.D2QU01.PHASE:$LMODO$SMODO -
CGI.LB.D2QU10.PHASE:$LMODO$SMODO -
CGI.LB.D2QU20.PHASE:$LMODO$SMODO -
CGI.LB.D2QV10.PHASE:$LMODO$SMODO -
CGI.LB.D2QV20.PHASE:$LMODO$SMODO -
CGI.LB.D2QV30.PHASE:$LMODO$SMODO -
CGI.LB.D2QX00.PHASE:$LMODO$SMODO -
CGI.LB.D2QX01.PHASE:$LMODO$SMODO -
CGI.LB.D2QY01.PHASE:$LMODO$SMODO -
CGI.LB.D2QY02.PHASE:$LMODO$SMODO -
CGI.LB.D2QY03.PHASE:$LMODO$SMODO -
CGI.LB.D2QY04.PHASE:$LMODO$SMODO -
CGI.LB.D2QY05.PHASE:$LMODO$SMODO -
CGI.LB.D2QY10.PHASE:$LMODO$SMODO -
CGI.LB.D2QY11.PHASE:$LMODO$SMODO -
CGI.LB.D2QY20.PHASE:$LMODO$SMODO -
CGI.LB.D2QY30.PHASE:$LMODO$SMODO -
CGI.LB.D2QZ00.PHASE:$LMODO$SMODO -
CGI.LB.D2Q000.PHASE:$LMODO$SMODO -
CGI.LB.D2Q100.PHASE:$LMODO$SMODO -
CGI.LB.D2Q101.PHASE:$LMODO$SMODO -
```

I N S T A L L A T I O N

4

4.6 INSTALLATION DES MODULES EXECUTABLES

22

```
CGI.LB.D2Q102.PHASE:$LMODO$SMODO -
CGI.LB.D2Q103.PHASE:$LMODO$SMODO -
CGI.LB.D2Q104.PHASE:$LMODO$SMODO -
CGI.LB.D2Q200.PHASE:$LMODO$SMODO -
===SEQ FOR TAB
CGI.LB.D2Q210.PHASE:$LMODO$SMODO -
===SEQ
CGI.LB.D2Q300.PHASE:$LMODO$SMODO -
CGI.LB.D2Q400.PHASE:$LMODO$SMODO -
CGI.LB.D2Q500.PHASE:$LMODO$SMODO -
CGI.LB.D2Q600.PHASE:$LMODO$SMODO -
CGI.LB.D2Q700.PHASE:$LMODO$SMODO -
CGI.LB.D2Q800.PHASE:$LMODO$SMODO -
CGI.LB.D2Q900.PHASE:$LMODO$SMODO -
CGI.LB.D2R000.PHASE:$LMODO$SMODO -
CGI.LB.D2R005.PHASE:$LMODO$SMODO -
CGI.LB.D2R100.PHASE:$LMODO$SMODO -
CGI.LB.D2R200.PHASE:$LMODO$SMODO -
CGI.LB.D2R400.PHASE:$LMODO$SMODO -
CGI.LB.D2R500.PHASE:$LMODO$SMODO -
CGI.LB.D2R600.PHASE:$LMODO$SMODO -
CGI.LB.D2R980.PHASE:$LMODO$SMODO -
CGI.LB.D2TPST.PHASE:$LMODO$SMODO -
CGI.LB.D2TPWS.PHASE:$LMODO$SMODO -
CGI.LB.D2UCTA.PHASE:$LMODO$SMODO -
CGI.LB.D2UCTR.PHASE:$LMODO$SMODO -
CGI.LB.D2UCTX.PHASE:$LMODO$SMODO -
TAPE=SYS004 REPLACE=Y SCAN=NO
ACCESS SUBLIB=$LMODO$SMODO
RENAME D2*.PHASE:$PRO.*.PHASE
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```

4.23. 4.7 CHARGEMENT DES FICHIERS SYSTEME

CHARGEMENT DES FICHIERS SYSTEME

Le module de JCL '3SYS' contient le JCL de définition et de chargement des fichiers système:

SC : squelette du générateur de programmes Batch
SG : squelette du générateur de programmes TP
SR : squelette du générateur COBOL
AEO : version de base des libellés d'erreurs VA PAC
SP : squelette variable (Extension PAF)
SF : squelette fixe (Extension PAF)
SS : squelette du générateur client/serveur
VGE : Dictionnaire Visualage

I N S T A L L A T I O N

4

4.7 CHARGEMENT DES FICHIERS SYSTEME

23

```

===MOD 3SYS
* $X JOB JNM=$PRFJ.3SYS,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.3SYS          ***** $PROD $PREL *****
*   INITIALISATION/CHARGEMENT DES FICHIERS SYSTEME PACBASE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- DEFINE FICHIERS SYSTEME PACBASE
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=D$PRO.$PRO.AE0.$SLIT
* $X SLI MEM=D$PRO.$PRO.VGE.$SLIT
* $X SLI MEM=D$PRO.$PRO.TCO.$SLIT
* $X SLI MEM=DF$PRO.$PRO.SC.$SLIT
* $X SLI MEM=DF$PRO.$PRO.SF.$SLIT
* $X SLI MEM=DF$PRO.$PRO.SG.$SLIT
* $X SLI MEM=DF$PRO.$PRO.SP.$SLIT
* $X SLI MEM=DF$PRO.$PRO.SR.$SLIT
* $X SLI MEM=DF$PRO.$PRO.SS.$SLIT
/*
/. JCLST02
* --- STEP 02 --- DECHARGEMENT SQUELETTE GENERATION BATCH
// DLBL IJSYSUC,'$VCAP',,VSAM
// ASSGN SYS004,$TAP1
// PAUSE MONTER BANDE $TAPEI SUR $TAP1
MTC REW,SYS004
MTC FSF,SYS004,3
// TLBL IN
// DLBL OUT,'$PXY..$PRO.$PRO.SC',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (3204) -
BLKSZ($BLKSC) NLBL)) -
OFILE (OUT) RUS
/*
/. JCLST03
* --- STEP 03 --- DECHARGEMENT SQUELETTE GENERATION DIALOGUE
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,4
// TLBL IN
// DLBL OUT,'$PXY..$PRO.$PRO.SG',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (4605) -
BLKSZ($BLKSG) NLBL)) -
OFILE (OUT) RUS
/*
/. JCLST04
* --- STEP 04 --- DECHARGEMENT SQUELETTE GENERATEUR COBOL
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,5
// TLBL IN
// DLBL OUT,'$PXY..$PRO.$PRO.SR',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (4605) -
BLKSZ($BLKSR) NLBL)) -
OFILE (OUT) RUS
/*
/. JCLST05
* --- STEP 05 --- DECHARGEMENT FICHER DES MESSAGES D'ERREUR
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,6
// TLBL IN
// DLBL OUT,'$PXY..SEQ.$PRO$PRO..AE0',,VSAM
// DLBL OUT2,'$PXY..$PRO.$PRO.AE',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (80) -
BLKSZ($BLKAE) NLBL)) -

```

```
        OFILE (OUT) RUS
REPRO IFILE(OUT) OFILE(OUT2) REP
/*
/. JCLST06
* --- STEP 06 --- DECHARGEMENT SQUELETTE EXTENSION PAF-PDM
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,8
// TLBL IN
// DLBL OUT,'$PX..$PRO.$PRO.SP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (4605) -
        BLKSZ($BLKSP)  NLBL)) -
        OFILE (OUT) RUS
/*
/. JCLST07
* --- STEP 07 --- DECHARGEMENT SQUELETTE FIXE EXTENSION PAF-PDM
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,9
// TLBL IN
// DLBL OUT,'$PX..$PRO.$PRO.SF',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (119) -
        BLKSZ($BLKSF)  NLBL)) -
        OFILE (OUT) RUS
/*
/. JCLST08
* --- STEP 08 --- DECHARGEMENT SQUELETTE CLIENT/SERVEUR
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,10
// TLBL IN
// DLBL OUT,'$PX..$PRO.$PRO.SS',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (4605) -
        BLKSZ($BLKSS)  NLBL)) -
        OFILE (OUT) RUS
/*
/. JCLST09
* --- STEP 09 --- DECHARGEMENT DICTIONNAIRE VISUALAGE
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,18
// TLBL IN
// DLBL OUT,'$PX..$PRO.$PRO.VGE',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (117) -
        BLKSZ($BLKVG)  NLBL)) -
        OFILE (OUT) RUS
/*
/. JCLST10
* --- STEP 10 --- DECHARGEMENT DICTIONNAIRE TEAMCONNECTION
// DLBL IJSYSUC,'$VCAP',,VSAM
MTC REW,SYS004
MTC FSF,SYS004,19
// TLBL IN
// DLBL OUT,'$PX..$PRO.$PRO.TCO',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE (IN ENV (PDEV (2400) RECFM(FB) RECSZ (117) -
        BLKSZ($BLKTC)  NLBL)) -
        OFILE (OUT) RUS
/*
// EXEC LISTLOG
/&
* $X EOJ
```

4.24. 4.8 CHARGEMENT DES FICHIERS UTILISATEUR

CHARGEMENT DES FICHIERS UTILISATEUR

Le module de JCL '3USR' contient le JCL d'initialisation de certains fichiers utilisateur:

AP : paramètres utilisateurs
PA : fichier de travail (PAF)
GS : schémas d'extraction (Extension PAF)

4.25. 4.9 CHARGEMENT DES LIBELLES D'ERREURS

CHARGEMENT LIBELLES D'ERREURS ET PARAMETRES UTILISATEUR

Le chargement s'effectue avec la procédure PARM.

TRES IMPORTANT - TRES IMPORTANT - TRES IMPORTANT - TRES IMPO

POUR FONCTIONNER, LE SYSTEME VA PAC DOIT IMPERATIVEMENT
CONTENIR AU MOINS LES PARAMETRES UTILISATEUR SUIVANTS :

- UN CODE UTILISATEUR
- LA CLE D'ACCES DU SYSTEME VA PAC

Les entrées fournies par défaut contiennent :

- une ligne NU : code utilisateur banalisé avec toutes les autorisations d'accès
aux bases

Code : TEST Mot de passe : à blanc

Ce code permet d'effectuer les tests d'installation et devra être supprimé
une fois entré le code utilisateur du responsable de la gestion de(s)
la(les) base(s) VA PAC .

- une ligne NK : RENSEIGNER ICI LA 'CLE D'ACCES' A VA PAC
COMMUNIQUEES AVEC LE COURRIER D'INSTALLATION

UTILISATION DE PACDESIGN - PARAMETRAGE DE LA METHODE

Entrer le paramétrage correspondant à la (aux) Méthode(s) utilisée(s) dans la Station de Travail.

Les mouvements de mise à jour se trouvent dans le jcl '9DSP'.

Les différentes versions de ces entités utilisateur sont sélectionnées par les variantes MER, YSM et SSA dans les commandes de l'utilitaire MM1JCL.

Si par erreur aucune ou plusieurs variantes ont été sélectionnées, il suffit de relancer le job d'installation JOB2 avec les cartes ===SELV appropriées et la carte ===SELM 9DSP

4.26. 5.1 CHARGEMENT BASE DE TESTS

CHARGEMENT DE LA BASE DE TESTS

Le job '5TPC' effectue le chargement de la sauvegarde de la base de tests fournie avec le produit. Celle-ci doit ensuite être restaurée par la procédure REST.

Cette base est fournie pour quelques tests initiaux de fonctionnement seulement.

```
===MOD 5TPC
* $X JOB JNM=$PRFJ.5TPC,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.5TPC          ***** $PROD $PREL *****
*          CHARGEMENT DE LA BASE DE TESTS VA PAC
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.PC.$SLIT
/*
/. JCLST02
* --- STEP 02 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// ASSGN SYS004,$TAP1
// PAUSE MONTER BANDE IMPLANTATION PACBASE $TAPEI SUR $TAP1
MTC REW,SYS004
MTC FSF,SYS004,7
// TLBL IN
// DLBL OUT,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO INFILE (IN ENV (PDEV (2400) RECFM(VB) RECSZ (155) -
BLKSZ($BLKPC) NLBL)) -
OUTFILE (OUT)
/*
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
```

4.27. 5.2 INIT COMMANDES D'EDITION-GENERATION

INITIALISATION DES COMMANDES D'EDITION-GENERATION

Cette initialisation s'effectue à l'aide de la procédure REAG avec une commande d'initialisation du fichier AG (AGI).

4.28. 5.3 INITIALISATION MODULE PEI

INITIALISATION ENVIRONNEMENT DE PRODUCTION

Pour la variante PEI seulement, l'initialisation s'effectue à l'aide de la procédure INPE suivie de la procédure RSPE.

4.29. 6.1 COMPLEMENT: INSTALLATION DICTIONNAIRE VISUAL

INSTALLATION DICTIONNAIRE VISUAL

Le chargement des Objets VISUAL sous forme d'Entités Utilisateur dans le Dictionnaire VA PAC s'effectue avec la procédure VINS.

Ce complément n'est à effectuer que pour les utilisateurs de la variante VISP.

4.30. 6.2.1 COMPLEMENT: ENVIRONNEMENT PAF

CREATION DU DICTIONNAIRE PAF

Le module de JCL '9PAF' contient les mouvements VA PAC à mettre en entrée de la procédure 'UPDT', afin de disposer dans votre base des entités standard du dictionnaire PAF. Ce dictionnaire servira de base aux développements de requêtes PAF.

4.31. 6.2.2 COMPLEMENT: EXTENSION PAF+

CREATION DES ENTITES UTILISATEUR DE L'EXTENSION PAF+

L'EXTENSION PAF+ exige pour son fonctionnement les éléments suivants :

- . Une Entité Utilisateur .PPTEx.
- . Les fichiers squelette SP et SF
- . Un fichier utilisateur GS contenant les Plans-Types d'Extraction (PTEx).

C'est en créant une occurrence de l'Entité Utilisateur que l'utilisateur pourra définir son Plan-Type d'Extraction.

Cette entité est rangée à l'installation dans le module de JCL '9XPD'. Celui-ci contient des mouvements batch destinés à être entrés dans la base VA PAC par la procédure batch UPDT de mise à jour du réseau. Lorsque la bibliothèque du réseau devant accueillir cette entité est choisie, modifier la ligne '*' en tête des mouvements, puis lancer la procédure UPDT.

Les squelettes PAF sont installés en standard:

- . Le squelette SP permet l'interprétation de l'occurrence d'Entité Utilisateur en termes de requêtes PAF.
- . Le squelette SF permet alors de générer un programme Cobol qui, une fois translaté au moyen du programme PAFP10, constituera un Extracteur Utilisateur ou une Macro-Commande appelée dans un Rapport.

Le fichier GS contient les Plans Type d'Extraction de l'utilisateur.

Un exemple de JCL d'exécution d'un Extracteur Utilisateur est fourni. C'est le module de JCL 'PAFX' qui est décrit au paragraphe qui suit.

I N S T A L L A T I O N

4

6.2.2 COMPLEMENT: EXTENSION PAF+

31

```

* EXEMPLE D'EXTRACTEUR UTILISATEUR PAF+
/* SUBSTITUTIONS A EFFECTUER
/* %EXTRAC = CODE EXTRACTEUR
/* %RECSZ = RECORDSIZE DU FICHIER EXTRAIT PAC7SQ
/* ENTREE : COMMANDE POUR L'EXTRACTION
/* COL. 2 : 'X'
/* COL. 3 : CODE CURSEUR PAF
/* COL. 7 : CODE OEU
/* COL. 15 : BORNE DEBUT
/* COL. 23 : BORNE FIN
/* COL. 31 : TYPE DEBUG (0 OU 1)
/* COL. 32 : NOMBRE DE BUFFERS PAF
/* *****
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DEFINE FICHIER TEMPORAIRE INPUT --- */
DEL ($XW..W$MODUL..INPUT) CL PURGE
DEF CL (NAME ($XW..W$MODUL..INPUT) -
MODEL($MODEL) -
REC (200 200) ) -
DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DEFINE FICHIER TEMPORAIRE SO --- */
DEL ($PXU..R$MODUL..SO) CL PURGE
DEF CL (NAME ($PXU..R$MODUL..SO) -
MODEL($MODEL) -
REC (100 100) -
RECSZ(436 436) -
RECFM(FB(436))) -
DATA (NAME ($PXU..R$MODUL..SO.D) )
/* --- DEFINE FICHIER TEMPORAIRE SQ --- */
DEL ($PXU..R$MODUL..SQ) CL PURGE
DEF CL (NAME ($PXU..R$MODUL..SQ) -
MODEL($MODEL) -
REC (100 100) -
RECSZ(%RECSZ %RECSZ) -
RECFM(FB(%RECSZ))) -
DATA (NAME ($PXU..R$MODUL..SQ.D) )
* $X SLI MEM=DFSYSYSPAF.$SLIT
/*
/. JCLST02
* --- STEP 02 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL OUT,'$XW..SYSPAF',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO OFILE (OUT) IFILE(SYSIPT) RUS
99999999999999
/*
* --- STEP 03 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER PASSWORDLIB
XTTTTCCCCCCCCDDDDDDDDFF1000100
/*
/. JCLST04
* --- STEP 04 --- %EXTRAC
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL SY8PAF,'$XW..SYSPAF',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7SO,'$PXU..R$MODUL..SO',,VSAM
// DLBL PAC7SQ,'$PXU..R$MODUL..SQ',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC %EXTRAC,SIZE=(AUTO,50K)
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO

```

I N S T A L L A T I O N
6.2.2 COMPLEMENT: EXTENSION PAF+

PAGE

157

4
31

```
DEL      ($XW..W$MODUL..INPUT) CL  
DEL      ($XW..SYSPAF)          CL  
/*
```

4.32. 6.3 COMPLEMENT: MODULE STATION DE TRAVAIL

CREATION DES ENTITES UTILISATEURS PACDESIGN

Le module de JCL '9DSG' contient les mouvements VA PAC à mettre en entrée de la procédure 'UPDT' avant d'utiliser la station de travail PACDESIGN.

Les différentes versions de ces entités utilisateur sont sélectionnées par les variantes MER, YSM et SSA dans les commandes de l'utilitaire MM1JCL.

Si par erreur aucune ou plusieurs variantes ont été sélectionnées, il suffit de relancer le job d'installation JOB2 avec les cartes ===SELV appropriées et la carte ===SELM 9DSG

4.33. 6.4 COMPLEMENT: PACBENCH QUALITY CONTROL

INSTALLATION PQC: REGLES COMPILEES ET ENTITE UTILISATEUR CREATION DU FICHIER DES REGLES DE QUALITE COMPILEES

Le module de JCL '9PQR' contient un jeu de règles de qualité standard compilées.

Ce fichier peut être transféré sur un fichier SAM-ESDS, de longueur fixe 80, afin d'être utilisé dans la procédure de suivi qualité PQCA (step 4, DLBL name PACQMF).

INSTALLATION ENTITE UTILISATEUR

L'option PERSONNALISATION du module PACBENCH QUALITY CONTROL nécessite pour son fonctionnement la présence dans la Base d'une Entité utilisateur.

Cette entité est fournie à l'installation dans le module de jcl '9PQU' sous la forme de mouvements batch destinés à être entrés dans la Base par la procédure batch UPDT de mise à jour du réseau.

Lorsque la bibliothèque du réseau devant accueillir cette entité est choisie, modifier le jcl '9PQU' sous éditeur en renseignant la ligne '*' en tête des mouvements, puis lancer la procédure UPDT.

4.34. 6.5 COMPLEMENT: INSTALLATION CARTES DE CONTROLE

INSTALLATION DE CARTES DE CONTROLE OPTIONNELLES

Le module de JCL '9INC' contient un flot de cartes de contrôle optionnelles pour la compilation de programmes batch ou TP, avec ou sans préprocesseur PACSQL, avec ou sans Documentation COBOL (DCOB). Si votre fichier des paramètres utilisateur (AP) ne contenait pas encore de telles cartes, vous pouvez les installer telles quelles. Dans le cas contraire, il serait prudent de vérifier les numéros de lignes afin qu'elles s'insèrent dans votre fichier sans interférer avec d'autres jeux de cartes.

I N S T A L L A T I O N

4

6.5 COMPLEMENT: INSTALLATION CARTES DE CONTROLE

34

```

===MOD 9INC
*
      INITIALISATION DES CARTES DE CONTROLE
NCD0  COMPILATION DE PROGRAMME COBOL BATCH (AVANT)
NCF0  COMPILATION DE PROGRAMME COBOL BATCH (APRES)
NCD1  COMPILATION DE PROGRAMME COBOL DIALOGUE (AVANT)
NCF1  COMPILATION DE PROGRAMME COBOL DIALOGUE (APRES)
NCD2  COMPILATION DE PROGRAMME BATCH AVEC PAF (AVANT)
NCF2  COMPILATION DE PROGRAMME BATCH AVEC PAF (APRES)
NCD3  COMPILATION DE PROGRAMME COBOL BATCH AVEC DCOB (AVANT)
NCF3  COMPILATION DE PROGRAMME COBOL BATCH AVEC DCOB (APRES)
NCD5  COMPILATION MAP BMS (AVANT)
NCF5  COMPILATION MAP BMS (APRES)
NCD7  COMPILATION DE PROGRAMME TP AVEC PAF (AVANT)
NCF7  COMPILATION DE PROGRAMME TP AVEC PAF (APRES)
NCD000/* COMPILATION DE PROGRAMME COBOL BATCH (AVANT)
NCD010* $X JOB JNM=$PRFJ.COB0,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'
NCD012* $X LST CLASS=$PCL,DISP=$PDL.$PWL
NCD019// JOB COB0 COMPIL COBOL BATCH (-/-/- LE -) $Y  PAGD -
NCD020* - : - $Y  SR -
NCD025// OPTION $OPT
NCD040* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
NCD080// OPTION CATAL
NCD082 PHASE -, * $Y  P -
NCD085 MODE RMODE(ANY)
NCD086// DLBL LUSER, '-', ,VSAM $Y  1 -
NCD087// LIBDEF PHASE,CATALOG=LUSER.- $Y  2 -
NCD089* --- STEP 1 --- IGYCRCTL
NCD090// EXEC IGYCRCTL,SIZE=IGYCRCTL
NCD097 CBL NOADV,APOST,RES,LIB,RENT,NOXREF,TRUNC(BIN)
NCF000/* COMPILATION DE PROGRAMME COBOL BATCH (APRES)
NCF010* --- STEP 2 --- LNKEDT
NCF030// EXEC LNKEDT
XNCF089/&
NCF099* $X EOJ
NCD100/* COMPILATION DE PROGRAMME COBOL DIALOGUE (AVANT)
NCD110* $X JOB JNM=$PRFJ.COB1,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'
NCD112* $X LST CLASS=$PCL,DISP=$PDL.$PWL
NCD119// JOB COB1 COMPIL COBOL DIALOGUE (-/-/- LE -)$Y  PAGD -
NCD120* - : - $Y  SR -
NCD125// OPTION $OPT
NCD140* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
NCD142// DLBL LUSER, '-', ,VSAM $Y  1 -
NCD144// LIBDEF PHASE,CATALOG=LUSER.- $Y  2 -
NCD150* --- STEP 1 --- DFHECP1$
NCD155// ASSGN SYS091,SYSPCH
NCD160// EXEC PROC=$$PACWK1,ASSGN=SYSPCH
NCD190// EXEC DFHECP1$
NCD197 CBL APOST,RES,LIB,RENT,NOXREF,TRUNC(BIN)
NCD199 CBL XOPTS(CICS,OPT,NOLIST)
NCF100/* COMPILATION DE PROGRAMME COBOL DIALOGUE (APRES)
NCF110* --- STEP 2 --- IGYCRCTL
NCF115 CLOSE SYSPCH,SYS091
NCF120// EXEC PROC=$$PACWK1,ASSGN=SYSIPT
NCF130// OPTION CATAL
NCF132 PHASE -, * $Y  P -
NCF134 INCLUDE DFHECI
NCF140// EXEC IGYCRCTL,SIZE=IGYCRCTL
NCF141/*
NCF150* --- STEP 3 --- LNKEDT
NCF190// EXEC LNKEDT
XNCF196/&
NCF197 CLOSE SYSIPT,SYSRDR
NCF199* $X EOJ
NCD200/* COMPILATION DE PROGRAMME BATCH AVEC PAF (AVANT)
NCD210* $X JOB JNM=$PRFJ.COB2,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'
NCD212* $X LST CLASS=$PCL,DISP=$PDL.$PWL
NCD219// JOB COB2 COMPIL COBOL AVEC PAF (-/-/- LE -)$Y  PAGD -
NCD220* - : - $Y  SR -
NCD225// OPTION $OPT
NCD240* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
NCD248// OPTION CATAL
NCD250 PHASE -, * $Y  P -
NCD255 MODE RMODE(ANY)

```

I N S T A L L A T I O N

4

6.5 COMPLEMENT: INSTALLATION CARTES DE CONTROLE

34

```

NCD260// DLBL LUSER, '- ', VSAM                $Y  1  -
NCD283// LIBDEF PHASE,CATALOG=LUSER.-          $Y  2  -
NCD290* $X SLI MEM=PAFB$PRO.$PFI..$SLIT
NCD293* $X DATA INPUT
NCF200/* COMPILATION DE PROGRAMME BATCH AVEC PAF (APRES)
NCF299* $X EOJ
NCD500/* COMPILATION MAP BMS (AVANT)
NCD510* $X JOB JNM=$PRFJ.MAP5,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'
NCD512* $X LST CLASS=$PCL,DISP=$PDL.$PWL
NCD519// JOB MAP5 COMPIL MAP BMS      (-/-/- LE -) $Y  PAGD -
NCD520* - : -                                $Y  SR  -
NCD525// OPTION $OPT
NCD552* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
NCD555// DLBL LUSER, '- ', VSAM                $Y  1  -
NCD563// LIBDEF PHASE,CATALOG=LUSER.-          $Y  2  -
NCD564// LIBDEF SOURCE,SEARCH=LUSER.-          $Y  2  -
NCD565* --- STEP 1 --- LIBR
NCD570// EXEC LIBR,PARM='AC S=LUSER.-;CA -.A R=Y' $Y  3P  -
NCF500/+ COMPILATION MAP BMS (APRES)
NCF505/*
NCF510* --- STEP 2 --- ASSEMBLAGE
NCF515// OPTION CATAL,SYSPARM='MAP',ALIGN
NCF520 PHASE -, *                             $Y  P  -
NCF530// EXEC ASSEMBLY
NCF535 PRINT ON,NOGEN
NCF540 COPY -                                  $Y  P  -
XNCF545 AIF ('&SYSPARM'(1,3) NE 'MAP').G565
NCF550 DS 0D
NCF552* CONSTANTES PACBASE DE LA MAP
NCF555 DC CL8'WORKING'
NCF560 DC CL5'- ',CL3'- ',CL8'- ',CL6'- ',CL8'- ' $Y  GADSU-
NCF565.G565 ANOP
NCF570 END
NCF575/*
NCF576* --- STEP 3 --- LNKEDT
NCF577// EXEC LNKEDT
NCF579/*
NCF580* --- STEP 4 --- LIBR
NCF582// DLBL IJSYSUC, '$VCAP', , VSAM
NCF585// DLBL LUSER, '- ', VSAM                $Y  1  -
NCF587// EXEC LIBR,PARM='AC S=LUSER.-;DEL -.A' $Y  2P  -
NCF588/*
XNCF589/&
NCF599* $X EOJ
NCD700/* COMPILATION DE PROGRAMME TP AVEC PAF (AVANT)
NCD710* $X JOB JNM=$PRFJ.COB7,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'
NCD712* $X LST CLASS=$PCL,DISP=$PDL.$PWL
NCD719// JOB COB7 COMPIL COBOL AVEC PAF (-/-/- LE -) $Y  PAGD -
NCD720* - : -                                $Y  SR  -
NCD725// OPTION $OPT
NCD740* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
NCD748// OPTION CATAL
NCD750 PHASE -, *                             $Y  P  -
NCD760// DLBL LUSER, '- '                      $Y  1  -
NCD783// LIBDEF PHASE,CATALOG=LUSER.-          $Y  2  -
NCD790* $X SLI MEM=PAFT$PRO.$PFI..$SLIT
NCD793* $X DATA INPUT
NCF700/* COMPILATION DE PROGRAMME TP AVEC PAF (APRES)
NCF799* $X EOJ
NCD300/* COMPILATION DE PROGRAMME COBOL BATCH AVEC DCOB (AVANT)
NCD310* $X JOB JNM=$PRFJ.COB3,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'
NCD312* $X LST CLASS=$PCL,DISP=$PDL.$PWL
NCD313* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
NCD319// JOB COB3 COMPIL COBOL AVEC DCOB(-/-/- LE -) $Y  PAGD -
NCD320* - : -                                $Y  SR  -
NCD325// OPTION $OPT
NCD343* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
NCD344// OPTION CATAL
NCD345 PHASE -, *                             $Y  P  -
NCD346 MODE RMODE(ANY)
NCD348// DLBL LUSER, '- ', VSAM                $Y  1  -
NCD349// LIBDEF PHASE,CATALOG=LUSER.-          $Y  2  -
NCD350* --- STEP 1 --- IGYCRCTL

```

I N S T A L L A T I O N

4

6.5 COMPLEMENT: INSTALLATION CARTES DE CONTROLE

34

```
NCD351// ASSGN SYS091,SYSLST
NCD352// EXEC PROC=$$PACWK2,ASSGN=SYSLST
NCD390// EXEC IGYCRCTL,SIZE=IGYCRCTL
NCD397 CBL NOADV,APOST,RES,LIB,RENT,NOXREF,TRUNC(BIN)
NCF300/* COMPILATION DE PROGRAMME COBOL BATCH AVEC DCOB (APRES)
NCF310* --- STEP 2 --- LNKEDT
NCF330// EXEC LNKEDT
NCF389/*
NCF390 CLOSE SYSLST,SYS091
NCF392// ASSGN SYS007,$LST2
NCF397* $X SLI MEM=DCOB$PRO.$PFI..$SLIT
XNCF398/&
NCF399* $X EOJ
/*
```

4.35. 6.6 COMPLEMENT: CATALOGAGE PROCEDURE DCOB

CATALOGAGE DE LA PROCEDURE DCOB

Le module de JCL 'DCOB' contient le source de la procédure DCOB\$PRO.\$PFI (procédure SLI), qui une fois cataloguée, pourra être mise en oeuvre par les cartes de contrôle optionnelles fournies (code d'appel '3').

I N S T A L L A T I O N

4

6.6 COMPLEMENT: CATALOGAGE PROCEDURE DCOB

35

```

===MOD DCOB
* $X JOB JNM=$PRFJ.DCOB,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.DCOB ***** $PROD $PREL *****
/* *****
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// DLBL $LSLI,'$SLIL',,VSAM
// EXEC LIBR,PARM='AC S=$LSLI$SLIS;CA DCOB$PRO$PFI..$SLIT R=Y'
/* *****
/* *****
*          COBOL DOCUMENTE
/* *****
/* LA PROCEDURE STANDARD DE JCL $$PACWK2, SI ELLE EST UTILISEE
/* DANS LE JOB DE COMPILATION AVEC LES PARAMETRES :
/*
/*          DLBL=IJSYSLS,ASSGN=SYSLST,OPT=PERM
/*
/* PERMET DE RECUPERER SUR DISQUE LE COMPTE-RENDU DE COMPILATION
/* DANS CE CAS, SI CE JOB PASSE DANS UNE PARTITION DIFFERENTE
/* DE CELLE DU JOB DE COMPILATION, IL SERA NECESSAIRE DE
/* REMPLACER '$$PACWK2' PAR 'PACWK2', 'PACWK2', ETC.
/* SELON QUE CELUI-CI AURA ETE EXECUTE DANS BG, F2, ETC.
/* LE FICHER DISQUE PAC8SY EST DE FORMAT UNDEFINED, BLKSZ=133
/* *****
* $P SLI MEM=DBJB$PRO.$PFI..$SLIT
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHER TEMPORAIRE INPUT --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL$) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
FIRST004007          COORDINATES 1ST COBOL LINE ON THE PAGE
WINDOW004064009     WINDOW (1ST LINE, BEHIND, 1ST COL)
NUMBR065            NUMBER OF LINES/PAGE
NULGG00105000001    NUM OF THE GENERATED LINE (COL, LTH, INCR)
SAUTP1              SKIP CHARACTERS AND CORRESPONDING SKIP
SAUTL+00
SAUTL 01
SAUTL002
SAUTL-03
/*
/. JCLST03
* --- STEP 03 --- PTUCR1
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAQYLC,'$XW..W$MODUL..LC',,VSAM,DISP=NEW
// EXEC PTUCR1,SIZE=AUTO,PARM='DCOB F01I'
/*
/. JCLST04
* --- STEP 04 --- PTUD10
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC8FA,'$XW..W$MODUL..FA',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC8FC,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC8LC,'$XW..W$MODUL..LC',,VSAM
// DLBL PAC8PG,'$XW..W$MODUL..PG',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC8RA,'$XW..W$MODUL..RA',,VSAM,DISP=(NEW,,DELETE)
/* DLBL PAC8SY          IF PAC8SY ON DISK (TO COMPLETE)
/* EXTENT SYS017,.....
/* ASSGN SYS017,DISK,VOL=.....,SHR
// EXEC PROC=$$PACWK2,DLBL=PAC8SY,ASSGN=SYS017 <--ALSO DISK
/* PAC8SY ON TAPE          SYS016 ASSIGNED
// ON $RC GT 4 GOTO STEPEND
// EXEC PTUD10,SIZE=AUTO

```

I N S T A L L A T I O N

4

6.6 COMPLEMENT: CATALOGAGE PROCEDURE DCOB

35

```

/*
// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- PTUD20
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC8FB,'$XW..W$MODUL..PG',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC8FE,'$XW..W$MODUL..FA',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC8FS,'$XW..W$MODUL..FS',,VSAM,DISP=(NEW,,DELETE)
// DLBL PAC8LC,'$XW..W$MODUL..LC',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 4 GOTO STEPEND
// EXEC PTUD20,SIZE=(AUTO,50K)
/*
// ON $RC<16 CONTINUE
/*
/. JCLST06
* --- STEP 06 --- SORT
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL SORTIN1,'$XW..W$MODUL..FS',,VSAM
// DLBL SORTOUT,'$XW..W$MODUL..KS',,VSAM,DISP=NEW
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC SORT,SIZE=(AUTO,50K)
SORT FIELDS=(1,15,CH,A)
RECORD TYPE=F,LENGTH=(100)
INPFIL VSAM
OUTFIL KSDS
/*
/. JCLST07
* --- STEP 07 --- PTUD30
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC8FA,'$XW..W$MODUL..KS',,VSAM
// DLBL PAC8FC,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC8LC,'$XW..W$MODUL..LC',,VSAM
// DLBL PAC8RA,'$XW..W$MODUL..RA',,VSAM,DISP=(,DELETE,KEEP)
/* DLBL PAC8SY IF PAC8SY ON DISK (TO COMPLETE)
/* EXTENT SYS016,.....
/* ASSGN SYS016,DISK,VOL=.....,SHR
// EXEC PROC=$$PACWK2,DLBL=PAC8SY,ASSGN=SYS016 <<--ALSO DISK
/* PAC8SY ON TAPE SYS015 ASSIGNED
// EXEC PTUD30,SIZE=AUTO
/. JCLST08
* --- STEP 08 --- PTUCR2
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAQYLC,'$XW..W$MODUL..LC',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAQYQN,'$XW..W$MODUL..QN',,VSAM
// EXEC PTUCR2,SIZE=AUTO
/. STEPEND
* --- STEP 09 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
/*
/+
/&
* $X EOJ

```

4.36. 7. MODIFICATION DU JCL DE CICS

MODIFICATION DU JCL DE LANCEMENT DE CICS

Une fois les tables CICS mises à jour et compilées et les fichiers correctement définis, le JCL de lancement de CICS doit être complété par le contenu du module '6JCI'.

Il contient :

- La DLBL de la bibliothèque de modules VA PAC -TP.
- La DLBL du catalogue VSAM utilisé pour VA PAC .
- Les DLBL des fichiers de chaque base: AN, AR, AJ, AG.
- Les DLBL des fichiers système: AE, AP, PA
- Les DLBL des fichiers complémentaires selon option: AB, AC pour PEI.

Il ne contient pas :

- La carte // LIBDEF PHASE,SEARCH=... qui existe déjà dans votre JCL.
Vous devez compléter celle-ci par l'appel de la Librairie .Sublib des modules VA PAC -TP.

```
===MOD 6JCI
/* *****
/*      --- MODIFICATIONS A APPORTER AU JCL DE CICS      *
* =====
*      AJOUTER $LMODO$SMODO  DANS LA  LIBDEF PHASE,SEARCH
* =====
/* *****
// DLBL $LMODO, '$MODO' , ,VSAM
// DLBL PACCT, '$VCAP' , ,VSAM
===SEQ FOR PEI
// DLBL $PRO$PFI.AB, '$PXU..$PRO$PFI.AB' , ,VSAM,CAT=PACCT
// DLBL $PRO$PFI.AC, '$PXU..$PRO$PFI.AC' , ,VSAM,CAT=PACCT
===SEQ
// DLBL $PRO$PRO.AE, '$PXY..$PRO$PRO.AE' , ,VSAM,CAT=PACCT
// DLBL $PRO$PFI.AG, '$PXU..$PRO$PFI.AG' , ,VSAM,CAT=PACCT
// DLBL $PRO$PFI.AJ, '$PXU..$PRO$PFI.AJ' , ,VSAM,CAT=PACCT
// DLBL $PRO$PFI.AN, '$PXU..$PRO$PFI.AN' , ,VSAM,CAT=PACCT
// DLBL $PRO$PRO.AP, '$PXY..$PRO$PRO.AP' , ,VSAM,CAT=PACCT
// DLBL $PRO$PFI.AR, '$PXU..$PRO$PFI.AR' , ,VSAM,CAT=PACCT
// DLBL $PRO$PRO.PA, '$PXY..$PRO$PRO.PA' , ,VSAM,CAT=PACCT
```

4.37. 8.1 COMPLEMENT: MODULE Pac/Impact

8.1. COMPLEMENT : MODULE Pac/Impact

Exécuter les job :
 'INFO'
 'INFP'
 'INFQ'

I N S T A L L A T I O N

4

8.1 COMPLEMENT: MODULE Pac/Impact

37

```

===SEQ FOR S2K
===MOD INFO
* $X JOB JNM=$PRFJ.INFO,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.INFO          ***** $PROD $PREL *****
*          INITIALISATION DU FICHIER FO
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHIER SAUVEGARDE FO --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWFO) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWFO) -
             MODEL($MODEL) -
             REC (1000 1000) -
             RECSZ(260 260) -
             RECFM(FB(260)) -
             ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWFO.D) )
/*
/. STEPEND
* --- STEP 02 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.FO.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

I N S T A L L A T I O N

4

8.1 COMPLEMENT: MODULE Pac/Impact

37

```

===SEQ FOR S2K
===MOD INFP
* $X JOB JNM=$PRFJ.INFP,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ' $PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.INFP          ***** $PROD $PREL *****
*
*          INITIALISATION DU FICHIER FP
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHIER TEMPORAIRE INPUT --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
            MODEL($MODEL) -
            REC (1 200) -
            ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE FICHIER PERMANENT FP --- */
DEL          ($PXU..$PRO.$PFI.FP) CL
DEF CL      (NAME ($PXU..$PRO.$PFI.FP) -
            NAL RUS VOL($VOLVWK) -
            KEYS (9 0) -
            REC (1000 1000) -
            RECSZ(9 9) -
            ) DATA (NAME ($PXU..$PRO.$PFI.FP.D) -
            ) INDEX (NAME ($PXU..$PRO.$PFI.FP.I) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
/*
/. JCLST03
* --- STEP 03 --- PAN205
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7FP,'$PXU..$PRO.$PFI.FP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GE 12 GOTO STEPEND
// EXEC PAN205,SIZE=(AUTO,50K)
/*
// ON $RC<16 CONTINUE
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

I N S T A L L A T I O N

4

8.1 COMPLEMENT: MODULE Pac/Impact

37

```

===SEQ FOR S2K
===MOD INFQ
* $X JOB JNM=$PRFJ.INFQ,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.INFQ          ***** $PROD $PREL *****
*                          REINITIALISATION DU FICHIER FQ
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHIER SAUVEGARDE FQ --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWFQ) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWFQ) -
             MODEL($MODEL) -
             REC (1000 1000) -
             RECSZ(100 100) -
             RECFM(FB(100)) -
             ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWFQ.D) )
/*
/. JCLST02
* --- STEP 02 --- PAN200
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7FQ,'$PXU..SEQ.$PRO$PFI..NEWFQ',,VSAM
// EXEC PAN200,SIZE=AUTO
/. STEPEND
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.FQ.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

4.38. 8.2 COMPLEMENT: MODULE PAC/TRANSFERT

8.2. COMPLEMENT : MODULE PAC/TRANSFER -----

Le module PAC/TRANSFER a pour but la gestion de données en mode batch grâce à l'utilisation de deux types de ressources :

. Des BIBLIOTHEQUES dans lesquelles seront stockés les programmes constitutifs du système, ainsi que les paramètres nécessaires à son fonctionnement :

Une bibliothèque de programmes batch,
Une bibliothèque de membres SLI (POWER),
Une bibliothèque de procédures VSE.

. Un FICHIER PERMANENT qui contient les paramètres utilisés pour effectuer les transferts.

CHARGEMENT DES FICHIERS UTILISATEUR

Le module de JCL '3IUUV' contient le JCL d'initialisation du fichier UV des paramètres de transfert.

4.39. 8.3 COMPLEMENT: PONT VA PAC / TEAMCONNECTION

8.3. COMPLEMENT : MODULE PONT VA PAC / TeamConnection

8.3.1. ALLOCATION DU FICHIER DES SESSIONS

Ce chargement est constitué du JOB qui comprend les étapes :

STEP1 : IDCAMS : Cette étape exécute le DELETE/DEFINE du fichier des sessions (TS).

STEP2 : IDCAMS : Initialisation du fichier des sessions (TS).

Ce traitement permet de créer sur le fichier des sessions un enregistrement particulier, sans lequel ce fichier est inutilisable.

8.3.2. INSTALLATION DU DICTIONNAIRE TEAMCONNECTION

Ce chargement est constitué du JOB

Il contient l'appel de la procédure VINS.

4.40. 9. LISTE DES PROGRAMMES ET FICHIERS INSTALLES

LISTE DES PROGRAMMES ET FICHIERS INSTALLES

Module de JCL '9INS'.

Vous obtenez :

- . La liste des programmes BATCH et TP avec les dates de compilation,
- . La liste des fichiers permanents système AE, SC, SF, SG, SP, SR, SS avec les dates de constitution.

Cette liste doit être conservée pour permettre de transmettre les références de l'installation au Support VisualAge Pacbase en cas de détection d'anomalies dans le fonctionnement de VA PAC .

I N S T A L L A T I O N

4

9. LISTE DES PROGRAMMES ET FICHIERS INSTALLEES

40

```

===MOD 9INS
* $X JOB JNM=$PRFJ.9INS,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.9INS          ***** $PROD $PREL *****
*      LISTE DES MODULES ET FICHIERS INSTALLEES
/* *****
/* ENTREE PACDTP :
/*
/* .SI UNE SELECTION DE PROGRAMMES EST SOUHAITEE, UNE LIGNE
/* PAR PROGRAMME SELECTIONNE DE FORMAT :
/* - CARACTERES 3-8 : CODE PROGRAMME SELECTIONNE
/*
/* .POUR OBTENIR LA LISTE DE TOUS LES PROGRAMMES, UNE SEULE
/* CARTE EN ENTREE DE FORMAT :
/* - CARACTERES 3-4 : RADICAL DU SYSTEME ($PRO)
/*
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DEFINE FICHIER TEMPORAIRE INPUT --- */
DEL      ($XW..W$MODUL..INPUT) CL PURGE
DEF CL (NAME ($XW..W$MODUL..INPUT) -
        MODEL($MODEL) -
        REC (200 200) ) -
        DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
$PRO
/*
/. JCLST03
* --- STEP 03 --- PACDTP
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7SC,'$PXY..$PRO.$PRO.SC',,VSAM
// DLBL PAC7SF,'$PXY..$PRO.$PRO.SF',,VSAM
// DLBL PAC7SG,'$PXY..$PRO.$PRO.SG',,VSAM
// DLBL PAC7SP,'$PXY..$PRO.$PRO.SP',,VSAM
// DLBL PAC7SR,'$PXY..$PRO.$PRO.SR',,VSAM
// DLBL PAC7SS,'$PXY..$PRO.$PRO.SS',,VSAM
// EXEC PACDTP,SIZE=AUTO
/*
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL      ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

4.41. TESTS D'INSTALLATION

TESTS DE L'INSTALLATION DE VA PACBASE

Ces tests comprennent plusieurs phases :

- . Tests d'utilisation de la Base,
- . Tests de gestion de la Base,
- . Tests des utilitaires d'extraction,

4.42. TESTS D'UTILISATION DE LA BASE

1. TESTS D'UTILISATION

Ces tests comprennent les étapes suivantes :

- . Tests d'utilisation en TP sous CICS,
- . Test de mise à jour batch,
- . Test d'édition et de génération de programmes.

Tests TP du Système :

Ouvrir les fichiers de la Base de test sous CICS.
Dérouler les enchaînements d'écrans.
Effectuer des mises à jour.

Test de mise à jour batch :

Exécuter le job '\$prfjUPDT' (Procédure UPDT).
Les fichiers de la Base doivent être fermés sous CICS.

Test d'édition-génération :

Exécuter le job '\$prfjGPRT' (Procédure GPRT).

Les fichiers de la Base doivent être fermés sous CICS si on lit le fichier des commandes d'édition et génération AG (mouvement +AG en entrée de GPRT).

*
* ENTREES DE LA PROCEDURE UPDT :
*...+...1...+...2...+...3...+...4...+...5...+...6...

*TEST CEN
0 TEST1 TEST
P5010 N TEST 10BL
P5010 1 M AA00 BB00
P5010 2 A 1 IWA10R

*
* ENTREES DE LA PROCEDURE GPRT :
*...+...1...+...2...+...3...+...4...+...5...+...6...

*TEST CEN
Z LCP
Z LCE
Z DCP TEST1

4.43. TESTS DE GESTION DE LA BASE

2. TESTS DE GESTION DE LA BASE VA PACBASE

Ces tests ont pour objet d'exécuter les procédures de gestion de la Base.

Ils comprennent les étapes suivantes, à exécuter dans l'ordre :

- . Archivage du journal créé lors des tests d'utilisation : exécuter le job '\$prfj.ARCH' donnant un fichier PJ(1).
- . Sauvegarde directe de la base : exécuter le job '\$prfj.SAVE' donnant un fichier PC(1).
- . Sauvegarde des commandes d'édition-génération : exécuter le job '\$prfj.SVAG' donnant un fichier PG.
- . Gestionnaire de bibliothèque : ajout/suppression de bibliothèque dans la Base : exécuter le job '\$prfj.MLIB' donnant un fichier PC(2).
- . Réorganisation de la sauvegarde séquentielle PC(2) de la Base : exécuter le job '\$prfj.REOR' donnant un fichier PC(3).
- . Réorganisation de la sauvegarde séquentielle PG du fichier des commandes d'édition-génération et restauration du fichier AG : exécuter le job '\$prfj.REAG'.
- . Restauration de la base à partir de l'archive PJ(1) et de la sauvegarde de la base PC(3) : exécuter le job '\$prfj.REST'.

Pour tous ces tests, les fichiers de la Base doivent être fermés sous CICS.

Après la restauration de la Base, refaire quelques tests rapides de fonctionnement en TP, après avoir réouvert les fichiers de la Base.

```
*
*      ENTREES DE LA PROCEDURE ARCH :
*.....1.....2.....3.....4.....5.....6....

S      1

*
*      ENTREES DE LA PROCEDURE MLIB :
*.....1.....2.....3.....4.....5.....6....

GR                                     BASE EXISTANTE -- PAS DE SIMULATION
C*AP5CEN                               CREATION BIBLIOTHEQUE AP5 (SOUS CEN)
*
*      ENTREES DE LA PROCEDURE REOR :
*.....1.....2.....3.....4.....5.....6....

V0010
BAP5
*
*      ENTREES DE LA PROCEDURE REAG :
*.....1.....2.....3.....4.....5.....6....

AG
*
*      ENTREES DE LA PROCEDURE REST :
*.....1.....2.....3.....4.....5.....6....

Y01000      REC                        123456789ABC
```

4.44. TESTS DES UTILITAIRES D'EXTRACTION

3. TESTS DES UTILITAIRES D'EXTRACTION

Ces tests ont pour objet d'exécuter les procédures d'extraction de la Base.

Ils comprennent les étapes suivantes, à exécuter dans l'ordre :

- . Extraction d'une bibliothèque du réseau sous forme de mouvements : exécuter le job '\$prfj.PACX' avec des entrées de type EXTR.
- . Extraction d'entités d'une bibliothèque : exécuter le job '\$prfj.PACX' avec des entrées de type EXLI.
- . Extraction de mouvements sélectionnés et/ou listes de mouvements sélectionnés à partir du journal archivé (PJ) : exécuter le job '\$prfj.PACX' avec des entrées de type EXPJ.

Pour tous ces tests, les fichiers de la Base peuvent être ouverts sous CICS.

Chacun des jobs peut enchaîner sur une mise à jour UPDT permettant de vérifier la validité des mouvements extraits.

I N S T A L L A T I O N
TESTS DES UTILITAIRES D'EXTRACTION

```
*
*      ENTREES DE LA PROCEDURE PACX DE TYPE EXLI :
*...+...1...+...2...+...3...+...4...+...5...+...6...

*TEST          CEN          EXLI1
*
*      ENTREES DE LA PROCEDURE PACX DE TYPE EXTR :
*...+...1...+...2...+...3...+...4...+...5...+...6...

*TEST          CEN          EXTR1
WLEX CEXCHOIX
WLEX CPBBMONI
WLEX UDXS
WLEX CSXS00
*
*      ENTREES DE LA PROCEDURE PACX DE TYPE EXPJ :
*...+...1...+...2...+...3...+...4...+...5...+...6...
*TEST          CEN          EXPJ1
JS
```

4.45. TESTS D ' UTILISATION : Pac/Impact

4. TESTS DU MODULE Pac/Impact

Ces tests ont pour objet d'exécuter les procédures du module Pac/Impact.

Pour effectuer les tests, il est conseillé d'établir une petite base par la procédure UXSR.

L'initialisation des fichiers FP et FQ doit être faite.

Les tests comprennent les étapes suivantes, à exécuter dans l'ordre:

- . détermination des points d'entrée, exécuter le job '\$prfj.ISEP'en indiquant les lignes de sélection, ainsi que le contexte.
- . détermination des points d'entrée complémentaires, exécuter le job '\$prfj.ISOS' en indiquant les lignes de sélection et le contexte.
- . Edition des points d'entrée, exécuter le job '\$prfj.IPEP'.
- . analyse d'impact, exécuter le job '\$prfj.IANA'. Pour les tests, une seule exécution de cette procédure peut suffire.
- . édition des résultats, exécuter le job '\$prfj.IPIA', en précisant le contexte et les éditions désirées.
- . édition des critères impactés, exécuter le job '\$prfj.IPFQ'.

4.46. TESTS D ' UTILISATION : PAC/TRANSFERT

5. TESTS DU MODULE PAC/TRANSFER

Ces tests ont pour objet d'exécuter les procédures du module de gestion du versioning PAC/TRANSFER (pour plus de détail, cf. Manuel 'GUIDE DE L'ADMINISTRATEUR, chapitre 'ADMINISTRATION DES VERSIONS').

Pour effectuer les tests, un archivage du journal de la base VA Pacbase doit être réalisé au préalable.

Les tests comprennent les étapes suivantes, à exécuter dans l'ordre :

- . Mise à jour des paramètres de transfert : exécuter le job '\$prfj.TRUP' en indiquant les paramètres de transfert désirés dans le flot des mouvements en entrée.
- . Compression du Journal archivé (ce test est facultatif) : exécuter le job '\$prfj.TRJC' en indiquant le nom du fichier du Journal archivé à compresser.
- . Création du fichier des transferts : exécuter : job '\$prfj.TRPF' en indiquant le code utilisateur son mot de passe, ainsi que le LOT de transfert à réaliser.
- . Préparation de l'Environnement DSMS (ce test facultatif n'est à réaliser que si la base VA Pac est sous contrôle DSMS) : exécuter le job '\$prfj.TRDU' en indiquant le LOT à traiter, puis effectuer la mise à jour batch de la base DSMS avec le premier flot des mouvements générés par la procédure DUPD.
- . Génération des mouvements de transfert : exécuter le job '\$prfj.TRRP' en indiquant le code utilisateur, son mot de passe en sélectionnant tous les LOTs et toutes les sessions à traiter.
- . Mise à jour de la base VA Pacbase par la procédure UPDT avec les mouvements de report générés.
- . Réinitialisation de l'environnement DSMS (ce test facultatif n'est à réaliser que si la base VA Pacbase est sous contrôle DSMS) : mettre à jour (en batch) la base DSMS avec le deuxième flot des mouvements générés par la procédure DUPD.

4.47. TESTS D ' UTILISATION : PONT VA PAC/VA SMALLTALK

6. PONT VISUALAGE PACBASE-VISUALAGE SMALLTALK

Pour tester les procédures du pont, il faut avoir à sa disposition, une base VA Pacbase de tests, ainsi qu'un fichier fourni par VisualAge (pour la procédure VUP1), compatible avec la base VA Pacbase de tests.

Les procédures doivent être exécutées dans l'ordre suivant :

- . VUP1 (JOB \$PRFJ.VUP1)
- . VUP2 (JOB \$PRFJ.VUP2)
- . VDWN (JOB \$PRFJ.VDWN)
- . VPUR (JOB \$PRFJ.VPUR)

5. REIMPLANTATION D'UNE SOUS-VERSION

5.1. REIMPLANTATION STANDARD

REIMPLANTATION DU SYSTEME

Une réimplantation du système VA PAC est à effectuer à la suite de la réception d'une sous-version apportant des corrections d'anomalies ou des développements ponctuels sur la version.

Cette sous-version, identifiée par un numéro, est constituée :

- . d'une cartouche (ou bande) d'implantation complète du produit,
- . d'une liste des anomalies corrigées,
- . éventuellement, d'une notice complétant la marche à suivre décrite dans le présent sous-chapitre pour la réimplantation.

D'une manière générale, seuls les fichiers du système et les bibliothèques de programmes sont touchés par une nouvelle version.

La réimplantation consiste donc, dans la majorité des cas, à exécuter les JOBS suivants :

1. JOB0, assurant la copie de la bande d'implantation (JOB contenu dans le fichier de JCL initial).
2. JOB2, assurant la création du JCL de réimplantation. Il exécute l'utilitaire MM1JCL avec les paramètres fournis lors de l'implantation, mais auxquels on ajoutera les lignes de sélection des modules de JCL suivantes :

```
===SELM 2PHA  
===SELM 3SYS
```

et éventuellement la ou les sélections supplémentaires mentionnées dans la notice complétant la présente marche à suivre.

Une fois les modules de JCL obtenus par ce JOB, ou récupérés de l'implantation ou la réimplantation précédente, la réimplantation se poursuit en exécutant les JOBS suivants.

3. Chargement des phases des programmes BATCH et/ou TP

Module 2PHA.

ATTENTION : vous devez, si telles sont les instructions données avec le produit, épurer ce JCL pour ne garder que les lignes de commande mentionnant les phases à réimplanter.

4. Rechargement des fichiers 'squelettes' de génération

Module 3SYS.

5. Rechargement du fichier libellés erreurs et documentation.

Procédure PARM.

Deux cas peuvent se présenter :

- Aucun paramètre utilisateur supplémentaire n'est à ajouter pour la réimplantation (pas de nouveaux utilisateurs, pas de nouvelles clés d'accès à VA PAC , etc.) et la sauvegarde paramètres (fichier PAC7CE) est valide : exécuter la procédure LOAE.
- Des paramètres utilisateur supplémentaires sont à ajouter pour la réimplantation : exécuter la procédure PARM avec la commande NRCHAR, suivie des mouvements de mise à jour.

REPRISE VA PACBASE 8.0, ..., 1.6
AVERTISSEMENT

PAGE

191

6
1

6. REPRISE VA PACBASE 8.0, ..., 1.6

6.1. AVERTISSEMENT

AVERTISSEMENT

Pour les utilisateurs des Modules STATION DE TRAVAIL, DSMS, et TABLES, il est nécessaire d'assurer la compatibilité de fonctionnement de ces différents Modules avec VisualAge Pacbase, en fonction des versions utilisées.

Cette nouvelle version de VisualAge Pacbase fonctionne avec :

- . STATION DE TRAVAIL nouvelle version
- . DSMS à partir de la version 8.0.2 compatible VA Pacbase 8.0.2
- . TABLES toutes versions.

ATTENTION : Pour les utilisateurs de TABLES 7.3 ou 8.0, il est nécessaire de disposer d'un programme PTA250 spécial, pour la procédure batch TABLES GETT. Ce programme est fourni sur demande.

	PAGE	193
REPRISE VA PACBASE 8.0, . . . , 1.6	6	
REPRISE VA PACBASE 8.0, 8.0.1	2	
OPERATIONS A EFFECTUER	1	

6.2. REPRISE VA PACBASE 8.0, 8.0.1

6.2.1. OPERATIONS A EFFECTUER

REPRISE VISUALAGE PACBASE 8.0, 8.0.1

OPERATIONS A EFFECTUER

La reprise permettant le passage d'un réseau PACBASE 8.0 ou 8.0.1 à un réseau exploitable par la nouvelle version de VisualAge Pacbase est composée des phases suivantes :

- . Réinstallation des paramètres utilisateur,
- . Réinstallation d'une Base VisualAge Pacbase,
- et éventuellement :
- . Réinstallation de l'environnement de production (PEI).

Des JCLs de reprise, à compléter par l'utilisateur, sont fournis pour l'enchaînement des différentes étapes à exécuter.

1-Réinstallation des paramètres utilisateur :

- . Sauvegarde des paramètres utilisateur donnant un fichier PE ancienne version (PARM ancienne version).
- . Reprise de la sauvegarde des paramètres utilisateur de la version 8.0 (Procédure PE80).

ATTENTION : Il est nécessaire de renseigner la CLE D'ACCES AU SYSTEME sur la ligne NK en entrée de la reprise.

- . Exécution de la procédure LOAE de la nouvelle version avec comme sauvegarde en entrée le fichier PE issu de la reprise et en entrée, l'utilisateur '*****' et la commande NRREST.
- . Pour l'utilisation de la Station de travail, exécution de la procédure PARM de la nouvelle version en incluant dans les entrées les mouvements de paramétrage de la méthode utilisée sur le site (voir au Chapitre "Installation", Sous-chapitre 4, Section 8 'Chargement libellés d'erreurs- documentation en ligne et paramètres utilisateur'). avec la commande NRCHAR.

Résultat obtenu : fichiers AE et AP contenant les paramètres utilisateurs ancienne version récupérés dans la nouvelle version et le paramétrage de la méthode.

	PAGE	194
REPRISE VA PACBASE 8.0, . . . , 1.6	6	
REPRISE VA PACBASE 8.0, 8.0.1	2	
OPERATIONS A EFFECTUER	1	

2-Réinstallation d'une Base VisualAge Pacbase :

- . Sauvegarde de la base donnant un fichier PC ancienne version.
- . Reprise du PC ainsi obtenu au nouveau format (Procédure PC80) donnant une sauvegarde PC nouvelle version.
- . Réorganisation (REOR nouvelle version) de la sauvegarde PC obtenue à l'étape précédente.
- . Initialisation du fichier journal
(Proc ARCH nouvelle version).
- . Restauration de la base à partir de la sauvegarde obtenue en sortie de la procédure de réorganisation (Proc REST nouvelle version).
- . Sauvegarde des commandes d'édition-génération donnant un fichier PG ancienne version.
- . Reprise des commandes d'édition-génération (RPPG) donnant un fichier PG nouvelle version.
- . Restauration des commandes d'édition-génération à partir de la sauvegarde précédente (Proc REAG nouvelle version).
- . Reprise de l'archive séquentielle (PJ80). Cette procédure est facultative. Elle permet de faire des extractions du journal sur les anciennes archives avec les nouveaux programmes traitant la date avec le siècle.

Résultat obtenu : Fichiers AJ, AN, AR et AG opérationnels dans la nouvelle version.

3-PEI : réinstallation environnement de production

- . Sauvegarde de l'environnement de production donnant un fichier PP ancienne version.
- . Reprise de la sauvegarde séquentielle (PP16)
La reprise consiste à ajouter le siècle à toutes les dates gérées par le module.
- . Restauration environnement de production à partir de la sauvegarde précédente (Proc RSPE nouvelle version).

Résultat obtenu : Fichiers AB et AC opérationnels dans la nouvelle version.

REPRISE VA PACBASE 8.0, ..., 1.6
 REPRISE VA PACBASE 8.0, 8.0.1
 PE80 : REPRISE PARAMETRES UTILISATEUR

6
 2
 2

6.2.2. PE80 : REPRISE PARAMETRES UTILISATEUR

```

===SEQ FOR ROLD
===MOD PE80
* $X JOB JNM=$PRFJ.PE80,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.PE80          ***** $PROD $PREL *****
*
*          REPRISE DES PARAMETRES UTILISATEUR
/* *****
/* SUBSTITUTIONS: fOLDPE PAR NOM PHYSIQUE PE ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHIER TEMPORAIRE INPUT --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE FICHIER SAUVEGARDE PE --- */
DEL          ($PXY..SEQ.$PRO$PRO..NEWPE) CL
DEF CL      (NAME ($PXY..SEQ.$PRO$PRO..NEWPE) -
             MODEL($MODEL) -
             REC (1000 1000) -
             RECSZ(80 80) -
             RECFM(FB(80)) -
             ) DATA (NAME ($PXY..SEQ.$PRO$PRO..NEWPE.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
XNK000<----- CLE D'ACCES AU SYSTEME -----$K
/*
/. JCLST03
* --- STEP 03 --- PTU902
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7IN,'fOLDPE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7OU,'$PXY..SEQ.$PRO$PRO..NEWPE',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU902,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PRO.PE.$SLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

REPRISE VA PACBASE 8.0, ..., 1.6
 REPRISE VA PACBASE 8.0, 8.0.1
 PC80 : REPRISE DE LA BASE

6
 2
 3

6.2.3. PC80 : REPRISE DE LA BASE

```

===SEQ FOR ROLD
===MOD PC80
* $X JOB JNM=$PRFJ.PC80,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.PC80          ***** $PROD $PREL *****
*                          REPRISE DE LA SAUVEGARDE DE LA BASE
/* *****
/* SUBSTITUTIONS: fOLDPC PAR NOM PHYSIQUE PC ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.PC.$SLIT
/*
/. JCLST02
* --- STEP 02 --- REPGDP
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MC,'fOLDPC',,VSAM
/* PAC7MC SUR BANDE: SYS011 ASSIGNEE
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7PC SUR BANDE: SYS012 ASSIGNEE
// EXEC REPGDP,SIZE=AUTO
/. STEPEND
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

REPRISE VA PACBASE 8.0, ..., 1.6

6

REPRISE VA PACBASE 8.0, 8.0.1

2

RPPG : REPRISE FICHER DES COMMANDES D'EDITION(AG)

4

6.2.4. RPPG : REPRISE FICHER DES COMMANDES D'EDITION(AG)

```

===SEQ FOR ROLD
===MOD RPPG
* $X JOB JNM=$PRFJ.RPPG,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.RPPG          ***** $PROD $PREL *****
*
*          REPRISE DU FICHER D'EDITION-GENERATION
/* *****
/* SUBSTITUTIONS: fOLDPG PAR NOM PHYSIQUE PG ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHER SAUVEGARDE PG --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPG) CL
DEF CL       (NAME ($PXU..SEQ.$PRO$PFI..NEWPG) -
              MODEL($MODEL) -
              REC (100 100) -
              RECSZ(150 150) -
              RECFM(FB(150)) -
              ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPG.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU908
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7IN,'fOLDPG',,VSAM
/* PAC7IN SUR BANDE: SYS011 ASSIGNEE
// DLBL PAC7OU,'$PXU..SEQ.$PRO$PFI..NEWPG',,VSAM
/* PAC7OU SUR BANDE: SYS012 ASSIGNEE
// ON $RC GT 0 GOTO $EOJ
// EXEC PTU908,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PG.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

	PAGE	198
REPRISE VA PACBASE 8.0, ..., 1.6		6
REPRISE VA PACBASE 8.0, 8.0.1		2
PJ80 : REPRISE DE JOURNAL ARCHIVE		5

6.2.5. PJ80 : REPRISE DE JOURNAL ARCHIVE

REPRISE DU JOURNAL ARCHIVE

Avec l'acquisition du Module DSMS, à partir de la version 8.0.2, l'utilisateur a la possibilité, lors de la reprise de sa version antérieure de DSMS, de modifier le Code Produit (dont le format passe de 1 à 3 caractères).

Cette possibilité est donc également prévue dans la reprise du Journal Archivé PACBASE d'une version antérieure.

ENTREE UTILISATEUR (Optionnelle)

```

-----
!Pos.! Lon.! Valeur ! Signification !
!-----!
! 1 ! 4 ! 'onnn' ! Changement du Code Produit DSMS !
! ! ! ! o = ancien Code Produit !
! ! ! ! nnn = nouveau Code Produit !
! ! ! ! (1, 2 ou 3 caractères) !
-----

```

Le couple 'onnn' peut être répété consécutivement sur la ligne autant de fois qu'il y a de Codes produits à modifier (20 fois maximum).

REMARQUE :

La validité de l'entrée n'est pas contrôlée. Il est donc important de bien vérifier la syntaxe de transposition avant exécution de la procédure qui n'édite pas de compte-rendu.

REPRISE VA PACBASE 8.0, ..., 1.6
 REPRISE VA PACBASE 8.0, 8.0.1
 PJ80 : REPRISE DE JOURNAL ARCHIVE

6
 2
 5

```

===SEQ FOR ROLD
===MOD PJ80
* $X JOB JNM=$PRFJ.PJ80,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ' $PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.PJ80          ***** $PROD $PREL *****
*
*          REPRISE DU JOURNAL ARCHIVE
/* *****
/* SUBSTITUTIONS: fOLDPJ PAR NOM PHYSIQUE PJ ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHER TEMPORAIRE INPUT --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE FICHER TEMPORAIRE PJ --- */
DEL          ($XW..W$MODUL..PJ) CL
DEF CL      (NAME ($XW..W$MODUL..PJ) -
             MODEL($MODEL) -
             REC ($RAJP $RAJS) -
             RECSZ(165 165) -
             RECFM(FB(165)) -
             ) DATA (NAME ($XW..W$MODUL..PJ.D) )
/* --- DELETE-DEFINE FICHER SAUVEGARDE PJ --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPJ) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ) -
             MODEL($MODEL) -
             REC ($RAJP $RAJS) -
             RECSZ(167 167) -
             RECFM(FB(167)) -
             ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
ONNN
/*
/. JCLST03
* --- STEP 03 --- PTU917
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7IN,'$OLDPJ',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7OU,'$XW..W$MODUL..PJ',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU917,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- REP2PJ
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7JP,'$PXU..SEQ.$PRO$PFI..NEWPJ',,VSAM
/* PAC7JP SUR BANDE: SYS011 ASSIGNEE
// DLBL PAC7PJ,'$XW..W$MODUL..PJ',,VSAM
/* PAC7PJ SUR BANDE: SYS012 ASSIGNEE
// ON $RC GT 0 GOTO STEPEND
// EXEC REP2PJ,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PJ.$SLIT

```

REPRISE VA PACBASE 8.0, ..., 1.6
REPRISE VA PACBASE 8.0, 8.0.1
PJ80 : REPRISE DE JOURNAL ARCHIVE

6
2
5

```
/*  
/. STEPEND  
* --- STEP 06 --- IDCAMS  
// DLBL IJSYSUC,'$VCAP',,VSAM  
// EXEC IDCAMS,SIZE=AUTO  
  DEL          ($XW..W$MODUL..INPUT) CL  
  DEL          ($XW..W$MODUL..PJ)    CL  
/*  
// EXEC LISTLOG  
/&  
* $X EOJ  
===SEQ
```

REPRISE VA PACBASE 8.0, ..., 1.6

6

REPRISE VA PACBASE 8.0, 8.0.1

2

PP16 : REPRISE ENVIRONNEMENT DE PRODUCTION

6

6.2.6. PP16 : REPRISE ENVIRONNEMENT DE PRODUCTION

```

===SEQ FOR ROLD
===MOD PP16
* $X JOB JNM=$PRFJ.PP16,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.PP16          ***** $PROD $PREL *****
*
*          REPRISE DE LA SAUVEGARDE PEI
/* *****
/* SUBSTITUTIONS: fOLDPP PAR NOM PHYSIQUE PP ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHER SAUVEGARDE PP --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPP) CL
DEF CL       (NAME ($PXU..SEQ.$PRO$PFI..NEWPP) -
              MODEL($MODEL) -
              REC (200 200) -
              RECSZ(110 110) -
              RECFM(FB(110)) -
              ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPP.D) )
/*
/. JCLST02
* --- STEP 02 --- PACR90
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7PE,'fOLDPP',,VSAM
/* PAC7PE SUR BANDE: SYS011 ASSIGNEE
// DLBL PAC7PS,'$PXU..SEQ.$PRO$PFI..NEWPP',,VSAM
/* PAC7PS SUR BANDE: SYS012 ASSIGNEE
// ON $RC GT 0 GOTO $EOJ
// EXEC PACR90,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PP.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

	PAGE	202
REPRISE VA PACBASE 8.0, ..., 1.6	6	
REPRISE VA PACBASE 802.02, ..., 1.6	3	
OPERATIONS A EFFECTUER	1	

6.3. REPRISE VA PACBASE 802.02, ..., 1.6

6.3.1. OPERATIONS A EFFECTUER

REPRISE VISUALAGE PACBASE 802.02, 1.2, 1.5, 1.6

OPERATIONS A EFFECTUER

L'installation de la nouvelle version de VisualAge Pacbase ne nécessite pas de reprise particulière des bases et fichiers utilisateur associés à l'exploitation de ces bases sauf pour le fichier des commandes d'édition-génération (AG), les fichiers PEI (AB, AC) et le journal archivé (PJ).

Lorsque l'installation a été faite, les opérations à effectuer consistent à sauvegarder les bases et fichiers associés par les procédures standard de la version à reprendre et à reprendre et restaurer ces mêmes bases et fichiers par les procédures standard fournies avec cette nouvelle version. Afin de profiter des nouveaux choix, il est intéressant d'inclure la procédure de réorganisation dans le processus de reprise.

Les Entités Utilisateur qui décrivent les entités spécifiques à la Station de Travail doivent être réintroduites dans la (les) base(s) par la procédure UPDT, une fois la (les) base(s) restaurée(s) dans la nouvelle version.

1-Réinstallation des paramètres utilisateur :

- . Sauvegarde des paramètres utilisateur donnant un fichier PE ancienne version (PARM 8.0.2,... 1.6).
- . Exécution de la procédure LOAE de la nouvelle version avec comme sauvegarde en entrée le fichier PE issu de la sauvegarde précédente, et en entrée, l'utilisateur '*****' et la commande NRREST.
- . Exécution de la procédure PARM avec, en entrée, le fichier mbparm contenant la nouvelle clé fournie avec le produit.
- . Pour l'utilisation de la Station de travail, exécution de la procédure PARM de la nouvelle version en incluant dans les entrées les mouvements de paramétrage de la méthode utilisée sur le site (voir au Chapitre "Installation", Sous-chapitre 4, Section 8 'Chargement libellés d'erreurs- documentation en ligne et paramètres utilisateur') avec la commande NRCHAR.

Résultat obtenu : fichiers AE et AP contenant les paramètres utilisateurs ancienne version récupérés dans la nouvelle version et le paramétrage de la méthode.

2-Réinstallation d'une Base VisualAge Pacbase :

	PAGE	203
REPRISE VA PACBASE 8.0, ..., 1.6	6	
REPRISE VA PACBASE 802.02, ..., 1.6	3	
OPERATIONS A EFFECTUER	1	

- . Sauvegarde de la base donnant un fichier PC ancienne version.
- . Initialisation du fichier journal
 - (Proc ARCH nouvelle version).
- . Restauration de la base à partir de la sauvegarde obtenue en sortie de la procédure de sauvegarde (Proc REST nouvelle version).
- . Sauvegarde des commandes d'édition-génération donnant un fichier PG ancienne version.
- . Reprise des commandes d'édition-génération (RPPG) donnant un fichier PG nouvelle version.
- . Restauration des commandes d'édition-génération à partir de la sauvegarde précédente (Proc REAG nouvelle version).
- . Reprise de l'archive séquentielle (PJ16). Cette procédure est facultative. Elle permet de faire des extractions du journal sur les anciennes archives avec les nouveaux programmes traitant la date avec le siècle.

Résultat obtenu : Fichiers AJ, AN, AR et AG opérationnels dans la nouvelle version.

3-PEI : réinstallation environnement de production

- . Sauvegarde de l'environnement de production donnant un fichier PP ancienne version.
- . Reprise de la sauvegarde séquentielle (PP16)
 - La reprise consiste à ajouter le siècle à toutes les dates gérées par le module.
- . Restauration environnement de production à partir de la sauvegarde précédente (Proc RSPE nouvelle version).

Résultat obtenu : Fichiers AB et AC opérationnels dans la nouvelle version.

REPRISE VA PACBASE 8.0, ..., 1.6

6

REPRISE VA PACBASE 802.02, ..., 1.6

3

RPPG : REPRISE FICHER DES COMMANDES D'EDITION(AG)

2

6.3.2. RPPG : REPRISE FICHER DES COMMANDES D'EDITION(AG)

```

===SEQ FOR ROLD
===MOD RPPG
* $X JOB JNM=$PRFJ.RPPG,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.RPPG          ***** $PROD $PREL *****
*
*          REPRISE DU FICHER D'EDITION-GENERATION
/* *****
/* SUBSTITUTIONS: fOLDPG PAR NOM PHYSIQUE PG ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHER SAUVEGARDE PG --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPG) CL
DEF CL       (NAME ($PXU..SEQ.$PRO$PFI..NEWPG) -
              MODEL($MODEL) -
              REC (100 100) -
              RECSZ(150 150) -
              RECFM(FB(150)) -
              ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPG.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU908
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7IN,'fOLDPG',,VSAM
/* PAC7IN SUR BANDE: SYS011 ASSIGNEE
// DLBL PAC7OU,'$PXU..SEQ.$PRO$PFI..NEWPG',,VSAM
/* PAC7OU SUR BANDE: SYS012 ASSIGNEE
// ON $RC GT 0 GOTO $EOJ
// EXEC PTU908,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PG.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

REPRISE VA PACBASE 8.0, ..., 1.6
 REPRISE VA PACBASE 802.02, ..., 1.6
 PJ16 : REPRISE DU JOURNAL

6
 3
 3

6.3.3. PJ16 : REPRISE DU JOURNAL

```

===SEQ FOR ROLD
===MOD PJ16
* $X JOB JNM=$PRFJ.PJ16,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.PJ16          ***** $PROD $PREL *****
*
*          REPRISE DU JOURNAL ARCHIVE
/* *****
/* SUBSTITUTIONS: fOLDPJ PAR NOM PHYSIQUE PJ ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHER SAUVEGARDE PJ --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPJ) CL
DEF CL       (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ) -
              MODEL($MODEL) -
              REC ($RAJP $RAJS) -
              RECSZ(167 167) -
              RECFM(FB(167)) -
              ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ.D) )
/*
/. JCLST02
* --- STEP 02 --- REP2PJ
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7JP,'$PXU..SEQ.$PRO$PFI..NEWPJ',,VSAM
/* PAC7JP SUR BANDE: SYS012 ASSIGNEE
// DLBL PAC7PJ,'fOLDPJ',,VSAM
/* PAC7PJ SUR BANDE: SYS011 ASSIGNEE
// ON $RC GT 0 GOTO $EOJ
// EXEC REP2PJ,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PJ.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

REPRISE VA PACBASE 8.0, ..., 1.6
 REPRISE VA PACBASE 802.02, ..., 1.6
 PP16 : REPRISE ENVIRONNEMENT DE PRODUCTION

6
 3
 4

6.3.4. PP16 : REPRISE ENVIRONNEMENT DE PRODUCTION

```

===SEQ FOR ROLD
===MOD PP16
* $X JOB JNM=$PRFJ.PP16,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.PP16          ***** $PROD $PREL *****
*
*          REPRISE DE LA SAUVEGARDE PEI
/* *****
/* SUBSTITUTIONS: fOLDPP PAR NOM PHYSIQUE PP ANCIENNE VERSION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE FICHER SAUVEGARDE PP --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPP) CL
DEF CL       (NAME ($PXU..SEQ.$PRO$PFI..NEWPP) -
              MODEL($MODEL) -
              REC (200 200) -
              RECSZ(110 110) -
              RECFM(FB(110)) -
              ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPP.D) )
/*
/. JCLST02
* --- STEP 02 --- PACR90
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7PE,'fOLDPP',,VSAM
/* PAC7PE SUR BANDE: SYS011 ASSIGNEE
// DLBL PAC7PS,'$PXU..SEQ.$PRO$PFI..NEWPP',,VSAM
/* PAC7PS SUR BANDE: SYS012 ASSIGNEE
// ON $RC GT 0 GOTO $EOJ
// EXEC PACR90,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PP.$SLIT
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```