



VisualAge Pacbase 2.5

**DSMS 2.5 IBM MVS/IMS
INSTALLATION & EXPLOITATION**

DEDIM000252F

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 (Août 1999)

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	9
2. LES COMPOSANTS DE DSMS	11
2.1. PRESENTATION GENERALE.....	12
2.2. LA BIBLIOTHEQUE DES PARAMETRES D'EXPLOITATION.....	13
2.3. LA BIBLIOTHEQUE DES PROGRAMMES TP.....	15
2.4. LA BIBLIOTHEQUE DES PROGRAMMES BATCH.....	17
2.5. BIBLIOTHEQUES SPECIFIQUES IMS.....	20
2.6. LES FICHIERS 'SYSTEME'.....	21
2.7. LES FICHIERS 'UTILISATEUR'.....	22
2.8. LA BIBLIOTHEQUE DES SOURCES POINTS D'ENTREE.....	25
3. ENVIRONNEMENT	26
3.1. ORGANISATION DES BASES DL/1.....	27
3.2. ENCOMBREMENT DU SYSTEME.....	30
4. LES PROCEDURES BATCH	31
4.1. PRESENTATION GENERALE.....	32
4.2. CLASSIFICATION DES PROCEDURES.....	33
4.3. ANOMALIES D'EXECUTION.....	35
5. ARCHIVAGE DU JOURNAL DES MISES A JOUR (DARC)	36
5.1. PRESENTATION GENERALE.....	37
5.2. ENTREES-TRAITEMENTS-RESULTATS.....	38
5.3. DESCRIPTION DES ETAPES.....	40
5.4. JCL DE LA PROCEDURE.....	42
6. IMPRESSION REQUETES ET DEMANDES D'EDITION (DPRT)	45
6.1. PRESENTATION GENERALE.....	46
6.2. ENTREES-TRAITEMENTS-RESULTATS.....	47
6.3. DESCRIPTION DES ETAPES.....	50
6.4. JCL DE LA PROCEDURE.....	51
7. RESTAURATION (DRST)	53
7.1. PRESENTATION GENERALE.....	54
7.2. ENTREES-TRAITEMENTS-RESULTATS.....	56
7.3. DESCRIPTION DES ETAPES.....	58
7.4. JCL DE LA PROCEDURE.....	61
8. SAUVEGARDE (DSAV)	64
8.1. PRESENTATION GENERALE.....	65
8.2. ENTREES-TRAITEMENTS-RESULTATS.....	66
8.3. DESCRIPTION DES ETAPES.....	67
8.4. JCL DE LA PROCEDURE.....	68
9. REORGANISATION DES REFERENCES CROISEES (DREO)	70
9.1. PRESENTATION GENERALE.....	71
9.2. ENTREES-TRAITEMENTS-RESULTATS.....	72
9.3. DESCRIPTION DES ETAPES.....	73
9.4. JCL DE LA PROCEDURE.....	75
10. EXTRACTION DU JOURNAL ARCHIVE VA PAC (DEXP)	78
10.1. PRESENTATION GENERALE.....	79
10.2. ENTREES-TRAITEMENTS-RESULTATS.....	80
10.3. DESCRIPTION DES ETAPES.....	81
10.4. JCL DE LA PROCEDURE.....	82

11. EXTRACTION D'ENTITES	(DEXT)	84
11.1. PRESENTATION GENERALE		85
11.2. ENTREES-TRAITEMENTS-RESULTATS		86
11.3. DESCRIPTION DES ETAPES		89
11.4. JCL DE LA PROCEDURE		90
12. EXTRACTION DE TABLES POUR LISTES EXTERNES	(DEXH)	92
12.1. PRESENTATION GENERALE		93
12.2. ENTREES-TRAITEMENTS-RESULTATS		94
12.3. DESCRIPTION DES ETAPES		95
12.4. JCL DE LA PROCEDURE		96
13. MISE A JOUR BATCH	(DUPT)	98
13.1. PRESENTATION GENERALE		99
13.2. ENTREES-TRAITEMENTS-RESULTATS		101
13.3. DESCRIPTION DES ETAPES		106
13.4. JCL DE LA PROCEDURE		107
14. INITIALISATION	(DINI)	109
14.1. PRESENTATION GENERALE		110
14.2. ENTREES-TRAITEMENTS-RESULTATS		111
14.3. DESCRIPTION DES ETAPES		113
14.4. JCL DE LA PROCEDURE		114
15. EXTRACTION DU JOURNAL POUR MISE A JOUR	(DXBJ)	115
15.1. PRESENTATION GENERALE		116
15.2. ENTREES-TRAITEMENTS-RESULTATS		117
15.3. DESCRIPTION DES ETAPES		118
15.4. JCL DE LA PROCEDURE		119
16. MODIFICATION DE CODES ET DE MOTS CLES	(DREN)	121
16.1. PRESENTATION GENERALE		122
16.2. ENTREES-TRAITEMENTS-RESULTATS		123
16.3. DESCRIPTION DES ETAPES		126
16.4. JCL DE LA PROCEDURE		127
17. PRE-PROCESSEUR DAF PROGRAMMES GENERES	(DPDF)	129
17.1. PRESENTATION GENERALE		130
17.2. ENTREES-TRAITEMENTS-RESULTATS		131
17.3. DESCRIPTION DES ETAPES		132
17.4. JCL DE LA PROCEDURE		133
18. MISE A JOUR BATCH A PARTIR DE TABLES DAF	(DUPD)	135
18.1. PRESENTATION GENERALE		136
18.2. ENTREES-TRAITEMENTS-RESULTATS		137
18.3. DESCRIPTION DES ETAPES		138
18.4. JCL DE LA PROCEDURE		140
19. INSTALLATION		142
19.1. PREPARATION ENVIRONNEMENT		143
19.2. PRESENTATION GENERALE		145
19.3. CARTOUCHE (OU BANDE) D'INSTALLATION		146
19.4. PREPARATION DE L'INSTALLATION		147
19.4.1. LE JCL INITIAL		149
19.4.2. INSTALLATION DU JCL COMPLET		154
19.4.3. OPTIONS PAR DEFAUT INSTALLATION		158
19.4.4. VARIANTES D'INSTALLATION		159
19.4.5. MODULES DE JCL		160
19.4.6. PARAMETRAGE DU JCL		161
19.4.7. SEPARATEURS DES MODULES DE JCL		165

19.5. DEROULEMENT DE L'INSTALLATION	166
19.5.1. CHARGEMENT DES PROGRAMMES BATCH ET TP	167
19.5.2. ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME.....	172
19.5.3. CHARGEMENT DES PROCEDURES BATCH.....	180
19.5.4. CHARGEMENT DE LA DBDLIB.....	182
19.5.5. CHARGEMENT DE LA PSBLIB	184
19.5.6. PREPARATION INITIALE DES FICHIERS.....	186
19.5.7. CHARGEMENT DES LIBELLES D'ERREUR.....	189
19.5.8. CHARGEMENT DES ACB.....	191
19.5.9. COMPLEMENT : SOURCES POINT ENTREE UTILISATEUR.....	193
19.5.10. CHARGEMENT DE LA BASE DE TEST DSMS.....	198
19.6. LISTE DES PROGRAMMES INSTALLES	200
19.7. TESTS D'UTILISATION	203
19.7.1. JCL TEST : DEXT.....	205
19.7.2. JCL TEST : DARC	206
19.7.3. JCL TEST : DSAV.....	207
19.7.4. JCL TEST : DRST.....	208
19.7.5. JCL TEST : DXBJ.....	209
19.8. REIMPLANTATION DU SYSTEME	210
20. REPRISE DSMS 8.0.1 EN DSMS 2.5 (DR80)	213
20.1. OPERATIONS A EFFECTUER	214
20.2. ENTREE UTILISATEUR	215
20.3. REPRISE DE LA SAUVEGARDE.....	216
20.4. JCL DE LA PROCEDURE	218
21. REPRISE DSMS 8.0.2 01/02 EN DSMS 2.5 (DR8Q)	220
21.1. OPERATIONS A EFFECTUER	221
21.2. REPRISE DE LA SAUVEGARDE.....	222
21.3. JCL DE LA PROCEDURE	223
22. REPRISE DSMS 1.2 OU 1.5 EN DSMS 2.5.....	224
22.1. PRESENTATION.....	225
22.2. PROCEDURE 'DR15' - DESCRIPTION DES ETAPES.....	226
22.3. PROCEDURE 'DR15' : JCL.....	227
22.4. PROCEDURE 'DR5J' - DESCRIPTION DES ETAPES	228
22.5. PROCEDURE 'DR5J' : JCL	229
23. REMPLACEMENT DES LOW-VALUES PAR DES BLANCS (DLVB)	230
23.1. DLVB : REMPLACEMENT DES LOW-VALUES PAR DES BLANCS.....	231
23.2. DLVB : PARAMETRES-DESCRIPTION DES ETAPES	232
23.3. DLVB : PROCEDURE D'EXECUTION	233

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
PREAMBULE

PAGE 9

1

1. PREAMBULE

AVERTISSEMENT

UTILISATION DU MANUEL

Ce manuel est destiné au responsable de l'installation et au gestionnaire de la base DSMS.

Il décrit les composants de DSMS, l'environnement système, les procédures batch, les consignes d'installation de la nouvelle version et les opérations à effectuer pour une réimplantation standard de versions de correction.

REMARQUES

Cette Version DSMS 2.5 implique une installation complète du lot technique : fichiers, programmes et procédures batch.

UTILISATEUR DISPOSANT D'UNE VERSION ANTERIEURE

Une fois l'installation effectuée, se reporter au chapitre concernant la reprise de la Version du site :

-Reprise DSMS 8.0.1 en DSMS 2.5

-Reprise DSMS 8.0.2 v01 ou v02 en DSMS 2.5

-Reprise DSMS 8.0.2 v01 ou v02 compatible VA Pac 8.0.1 en DSMS 2.5

-Reprise DSMS 1.2 ou 1.5 en DSMS 2.5

et suivre rigoureusement les instructions indiquées afin d'assurer la compatibilité ascendante entre la nouvelle version et la version installée précédemment.

VisualAge Pacbase - Manuel d'Exploitation	PAGE	11
DSMS - EXPLOITATION & INSTALLATION		
LES COMPOSANTS DE DSMS		2

2. LES COMPOSANTS DE DSMS

2.1. PRESENTATION GENERALE

PRESENTATION GENERALE

Le module DSMS gère des données en mode conversationnel et en mode batch.
Les ressources utilisées pour son fonctionnement sont de trois types :

- Des bibliothèques dans lesquelles sont stockés les programmes constitutifs du module :
 - . une bibliothèque de programmes TP,
 - . une bibliothèque de programmes batch.

- Des fichiers permanents qui matérialisent les données manipulées par les programmes définis précédemment :
 - . un fichier 'système' contenant les libellés d'erreur et la documentation automatique du module,
 - . des fichiers 'utilisateur' contenant les informations de l'administrateur et des utilisateurs.

- Une bibliothèque contenant les paramètres d'exploitation

REMARQUE : Le module DSMS peut être installé indépendamment des autres outils VisualAge Pacbase. Sa mise en place et son exploitation sont décrites dans le présent manuel.

ATTENTION: Pour la plateforme IMS, le fichier des éléments VA Pac (DC-D3) est défini lors de l'installation VisualAge Pacbase.

Pour plus de détails sur le fonctionnement du module lui-même, se reporter au Manuel de Référence DSMS.

2.2. LA BIBLIOTHEQUE DES PARAMETRES D'EXPLOITATION

LA BIBLIOTHEQUE DE PARAMETRES : SY

La taille nécessaire est d'environ 9 pistes (disque 3390).

C'est un fichier PDS qui contient les entrées des utilitaires utilisés lors de l'installation et dans les procédures batch d'exploitation, ainsi que les sources des DBDs et PSBs

.Les sources DL/1:

```
+-----+-----+
! Membres ! Contenu !
+-----+-----+
! PACDwwxx ! DBD de la base ww, suffixe xx !
! Ppppppyy ! PSB du programme batch, suffixe yy !
! rr00zz ! rr: radical du PSB du moniteur TP zz !
+-----+-----+

+-----+-----+
! Membres ! Contenu : DELETE/DEFINE du fichier !
+-----+-----+
! DFxxyyDA ! données DSMS (DA) !
! DFxxyyD1 ! !
! DLxxyyDJ ! journal DSMS (DJ) !
! DFxxyyDX ! références croisées (DX) !
! DFxxxxDE ! libellés d'erreurs et doc.automatique (DE) !
! DFxxxxDZ ! base des SPA (DZ) !
! DFxxxxDF ! fichier de travail DAF TP !
! DFxxxxF1 ! !
+-----+-----+
```

Voir les caractéristiques physiques des fichiers dans les sous-chapitres qui suivent.

xx = ROOT Radical du système,
yy = FILE Numéro de base de DSMS utilisateur.

Les informations concernant le catalogue utilisé, les disques, l'encombrement, etc., sont initialisées en fonction du paramétrage initial de l'implantation et doivent être modifiées par le responsable de l'exploitation du module en fonction des besoins réels des utilisateurs du module.

.Les VERIFY,REPRO et LISTCAT des fichiers DSMS:

- dans les membres VERIFff (ff = DA, DC, DJ, DX,...), on trouve l'ordre VERIFY PACDff pour chacun des fichiers DSMS,
- dans les membres REPROff (ff = DZ, DF), on trouve l'ordre REPRO INFILE (INzz) OUTFILE (OUTzz).
- dans le membre LIxxyyDJ, on trouve le LISTCAT du fichier journal DJ.

.Les DFSVSAMP:

- dans les membres DFSVSAMn(n=8,9 ou M), on trouve les entrées des cartes de contrôle de la 'bufférisation' VSAM. Ces cartes sont initialisées à l'implantation du système avec des valeurs usuelles, mais leur gestion est à la charge du responsable système du produit.

Les paramètres définis sous le nom DFSVSAMn avec n=M, permettent d'optimiser les temps de mise à jour batch lors de l'exécution de la procédure correspondante.

.Les enregistrements pour initialiser DZ et DF:

- dans les membres LDxxyyDF LDxxyyDZ.

.Les macro-instructions APPLCTN et TRANSACT

- Sous le nom PACSCTRL on trouve l'ensemble des macros- instructions devant être définies dans la région de contrôle IMS. Ce PDS est destiné à l'équipe système.

REMARQUE

Toute modification des caractéristiques des fichiers doit impérativement être effectuée dans cette bibliothèque des paramètres.

2.3. LA BIBLIOTHEQUE DES PROGRAMMES TP

LA BIBLIOTHEQUE DES PROGRAMMES TP

Taille : environ 70 pistes pour un disque 3390

! CODE	! CHOIX CORRESPONDANT	!
! PROGRAMME	! Commentaires	!
! xxCHOI	! Sous-pgm décodification choix	!
! xxCUAM	! Sous-pgm ctrl supplém. sur xx00B1	!
! xxCUEV	! Sous-pgm ctrl supplém. sur xx00E1	!
! xxCUMQ	! Sous-pgm ctrl supplém. sur xx00Q6	!
! xxCURQ	! Sous-pgm ctrl supplém. sur xx00Q1	!
! xxCUSI	! Sous-pgm ctrl supplém. sur xx00SI	!
! xxR980	! Mise en forme du message 3270	!
! xx0099	! Moniteur DSMS	!
! xx00ZZ	! Moniteur DSMS	!
! xx00AA	! Ecran initial	!
! xx00AB	! Map Abend	!
! xx00BA	! HC	!
! xx00B1	! C	!
! xx00B2	! C C	!
! xx00B3	! C Q	!
! xx00B4	! C M	!
! xx00B5	! XS	!
! xx00EA	! HE	!
! xx00E1	! E	!
! xx00E2	! C D E DN/DT	!
! xx00E3	! C F E FN/FT	!
! xx00E4	! C T E T	!
! xx00E5	! LCE_	!
! xx00E6	! C S E S	!
! xx00FA	! HPF	!
! xx00FB	! HSC	!
! xx00HE	! Fonction Help	!
! xx00JO	! JO	!
! xx00KA	! HK	!
! xx00K1	! LGK_ LAK_	!
! xx00K2	! LPK_	!
! xx00K3	! WS WU	!
! xx00LE	! LDE_ LNC_ LSE_ LDC LNC	!
! xx00LS	! LIE*....	!
! xx00MA	! H (menu général)	!
! xx00PA	! HP	!
! xx00P1	! PL	!
! xx00QA	! HQ	!
! xx00QB	! Q C DD	!
! xx00QC	! R CD	!

! CODE	! CHOIX CORRESPONDANT	!
! PROGRAMME	! Commentaires	!
! xx00Q1	! Q	!
! xx00Q2	! Q D	!
! xx00Q3	! LCQ	!
! xx00Q4	! LVQ	!
! xx00Q5	! LJQ	!
! xx00Q6	! R	!
! xx00Q7	! R L	!
! xx00Q8	! R C	!
! xx00Q9	! LCR	!
! xx00SA	! HS	!
! xx00SI	! S * . . . U	!
! xx00S1	! S * . . .	!
! xx00S3	! S * . . . V	!
! xx00S4	! S * . . . C	!
! xx00S5	! S * . . . LC	!
! xx00S6	! S * . . . G	!
! xx00S7	! LSS	!
! xx00S8	! LNS LCS	!
! xx00S9	! S * . . . LV	!
! xx00TA	! HT	!
! xx00TT	! TUP	!
! xx00TU	! TRA	!
! xx00TV	! TLA	!
! xx00TW	! TPH	!
! xx00TX	! TUG	!
! xx00TY	! TUS	!
! xx00TZ	! TOP	!
! xx00T1	! TST	!
! xx00T2	! TSU	!
! xx00T3	! TGR	!
! xx00T4	! TPR	!
! xx00T5	! TRE	!
! xx00T6	! TTY	!
! xx00T7	! TUD	!
! xx00T8	! TVE	!
! xx00T9	! TAT	!
! xx00UD	! Upload/Download traitement texte	!

! CODE	!	!
! PROGRAMME	! Commentaires	!
! PACSECB	! Interface système de sécurité	!
! PACX10	! Prise en compte des minuscules	!
! PACXAB	! Gestion des abends	!
! MVSJOB	! Fonction 'JOB' MVS	!

REMARQUE

xx est le préfixe des programmes, correspondant au paramètre ROOT, 2 premiers caractères du code transaction choisi.

2.4. LA BIBLIOTHEQUE DES PROGRAMMES BATCH

LA BIBLIOTHEQUE DES PROGRAMMES BATCH : MBR8

Taille : environ 55 pistes pour un disque 3370.

! CODE	! PROC.	! Commentaire	!
! DAFD10	! DPDF	! Pré-processing DAF	!
! PACSECB	!	! Interface système de sécurité	!
! PDCHOI	! DUPT	! Sous-programme décodification choix	!
! PDSA10	! DPRT	! Sous-programme édition chaîne DPRT	!
! PDSB	! -	! Moniteur enchaînement DPRT (français)	!
! PDSBE	! -	! Moniteur enchaînement DPRT (anglais)	!
! PDSBAS	! DSAV	! Vérifie intégrité des données	!
! PDSCAM	! DUPT	! Ss-pgm ctls supplémentaires pour PDSUB1	!
! PDSCEV	! -	! Ss-pgm ctls supplémentaires pour PDSUE1	!
! PDSCMQ	! -	! Ss-pgm ctls supplémentaires pour PDSUQ6	!
! PDSCRQ	! -	! Ss-pgm ctls supplémentaires pour PDSUQ1	!
! PDSCSI	! -	! Ss-pgm ctls supplémentaires pour PDSUS1	!
! PDTPDF	!	! Sous-programme d'extraction DAF	!
! PDSEX	! DEXT	! Moniteur enchaînement DEXT (français)	!
! PDSEXE	! DEXT	! Moniteur enchaînement DEXT (anglais)	!
! PDSE90	! DPRT	! Sous-programme édition chaîne DPRT	!
! PDSERQ	! DPRT	! Sous-pgm édition (STATIQUE avec PDSE90)	!
! PDSFAC	!	! Sous-programme d'accès aux fichiers	!
! PDSINI	! DINI	! Initialise les fichiers DSMS	!
! PDSJMS	! DREN	! Changement des codes sur le journal	!
! PDSLVB	! DLVB	! Remplacement des low-value par des	!
!	! -	! blancs dans la sauvegarde BB	!
! PDSMS	! DREN	! Moniteur de remplacement des codes	!
!	! -	! table, mots clés et codes site	!
! PDSMSE	! -	! idem PDSMS en anglais	!
! PDSRCT	! DREN	! Contrôle des mouvements en entrée	!
! PDSRFU	! -	! Tri fusion	!
! PDSRMS	! -	! Changement des codes sur la sauvegarde	!
! PDSRQ0	! DPRT	! Analyse préliminaire	!
! PDSRQ1	! -	! Sélection des requêtes	!
! PDSRQ2	! -	! Mise en forme des éléments	!
! PDSRQ3	! -	! Extraction et éditions	!
! PDSR10	! DREO	! Réorganisation	!
! PDSR20	! -	!	!
! PDSR30	! -	!	!
! PDSR40	! -	!	!
! PDSUAA	! DUPT	! Sous-programme	!
! PDSUB1	! -	!	!
! PDSUB2	! -	!	!
! PDSUB3	! -	!	!

! CODE	! PROC.	! Commentaire
! PDSUB4	! DUPT	! -
! PDSUE1	! -	! -
! PDSUE2	! -	! -
! PDSUE3	! -	! Sous-programme
! PDSUK1	! -	! -
! PDSUP0	! -	! Moniteur MAJ batch
! PDSUP1	! -	! Sous-programme
! PDSUQ1	! -	! -
! PDSUQ2	! -	! -
! PDSUQ5	! -	! -
! PDSUQ6	! -	! -
! PDSUQ7	! -	! -
! PDSUQ8	! -	! -
! PDSUS1	! -	! -
! PDSUS1	! -	! -
! PDSUS3	! -	! -
! PDSUS4	! -	! -
! PDSUS6	! -	! -
! PDSUTT	! -	! -
! PDSUTV	! -	! -
! PDSUTW	! -	! -
! PDSUTX	! -	! -
! PDSUTY	! -	! -
! PDSUTZ	! -	! -
! PDSUT1	! -	! -
! PDSUT2	! -	! -
! PDSUT3	! -	! -
! PDSUT4	! -	! -
! PDSUT5	! -	! -
! PDSUT6	! -	! -
! PDSUT7	! -	! -
! PDSUT8	! -	! -
! PDSUT9	! -	! -
! PDSXCT	! DEXT	! Sous-programme validité entrées
! PDSXDT	! DINS	! Liste les programmes installés
! PDSXST	! DEXT	! Sous-programme tri
! PDSXTH	! DEXH	! Extraction de tables pour listes ext.
! PDSXTR	! -	! - extraction
! PDS300	! DARC	! Archive désactive le journal
! PDS320	! -	! Réinitialise le journal
! PDS380	! DRST	! Vérifie le journal
! PDS400	! -	! Restauration/initialisation fichiers
! PDS450	! -	! Réapplique le journal archivé
! PDS500	! DSAV	! Sauvegarde données/éléments/références
! PDS600	! DEXP	! Extraction du journal VA Pac
! PDS610	! -	! -
! REP2PJ	! DEXQ	! Extraction du journal VA Pac < 2.0

! CODE	! PROC.	! Commentaire	!
! PDS700	! DXBJ	! Extraction de mouvements du journal	!
! PDS900	! DUPD	! Mise à jour base DSMS - DAF	!
! PTU001	! Toutes	! Recopie des entrées sur fichier disque	!
! PDSR8B	! DR80	! Reprise base DSMS 8.0 ou 8.0.1	!
! PDSR8C	!	! ATTENTION : VA PAC DOIT ETRE	!
!	!	! PREALABLEMENT BASCULE EN 8.0.2.	!
! PDSR8Q	! DR8Q	! Reprise base DSMS 8.0.2 01 ou 02	!
! PDSR8R	!	! (Reprise des Requêtes)	!
! PDSR15	! DR15	! Reprise de base DSMS 1.2 ou 1.5	!
! PDSR5J	! DR5J	! Reprise journal archivé DSMS 1.5	!
! PDSV10	! DLDE	! Chargement base des libellés d'erreur	!
! PDSV20	! DLDZ	! Chargement base des S.P.A	!
! PTV090	! LDDF	! Chargement base fichier DAF	!

EXTENSION INTERFACE SYSTEMES DE SECURITE

Des sous-programmes particuliers assurent la connexion de DSMS avec le système de sécurité du site, lorsque cette extension a été choisie.

Pour RACF, le sous-programme PACSECU8 doit être installé dans une bibliothèque autorisée, par copie du module se trouvant dans la bibliothèque des modules batch (PACD.MBR8).

! PROGR.	! Renommé	! Système de sécurité	!
! PACSECRA	! PACSECU8	! RACF	!
! PACTSS	!	! TOPSECRET Batch	!

Pour l'exploitation de cette extension, se reporter au Manuel de Référence INTERFACE SYSTEMES DE SECURITE.

2.5. BIBLIOTHEQUES SPECIFIQUES IMS

BIBLIOTHEQUES SPECIFIQUES IMS:

Elles sont au nombre de deux

- . Une bibliothèque des DBDs (DBDLIB),
- . Une bibliothèque des PSBs (PSBLIB),

Dans le cas où l'utilisateur n'utilise pas des librairies systèmes existant sur le site, il devra les créer suivant les caractéristiques suivantes:

- . Organisation : PDS
- . DSN : au choix à l'implantation
- . DCB : RECFM=U,BLKSIZE=6144
- . Taille : voir chapitre ENVIRONNEMENT

2.6. LES FICHIERS 'SYSTEME'

LES FICHIERS 'SYSTEME'

Ils constituent le système proprement dit. Ils ne sont pas touchés par les manipulations quotidiennes et doivent faire l'objet d'un rechargement lors d'une réimplantation.

Ce sont les BIBLIOTHEQUES décrites dans les sous-chapitres précédents :

- . la bibliothèque des modules exécutables TP MTR8,
- . la bibliothèque des modules exécutables batch MBR8,
- . la bibliothèque des paramètres d'exploitation SY,

le fichier contenant les MESSAGES D'ERREURS et la DOCUMENTATION AUTOMATIQUE du module DSMS (DE) :

```
.Taille          : Environ 30000 enregistrements  
.Organisation   : base DL/1 HISAM  
.Lng SEGM       : 90  
.Lng RECORD     : 98  
.Clé            : 17 (position 0)  
.DSN            : &INDEX..&ROOT.&ROOT.DE  
.DBD            : PACDDExx
```

2.7. LES FICHIERS 'UTILISATEUR'

LES FICHIERS 'UTILISATEUR'

Ils contiennent les informations de l'utilisateur qui sont gérées par le module.

Les six premiers (4+2 index) constituent les données directement gérées par le module. Ce sont :

.Fichier des données DSMS (DA)

.Organisation : Base DL/1 HIDAM/VSAM
.Lng SEGM. : mini 80, maxi 350
.Lng RECORD : 4096 octets
.DSN : &INDEX..&ROOT.&FILE.DA
.DBD : PACDDAxx
.Encombrement : suivant longueur des enregistrements

.Index primaire des données (D1)

.Organisation : Index 'DA' INDEX/VSAM
.Lng SEGM. : 40 octets
.Lng RECORD : 59 octets
.DSN : &INDEX..&ROOT.&FILE.D1
.DBD : PACDD1xx
.Encombrement : 22 enreg. par CI de 1024

.Fichier des éléments VisualAge Pacbase (DC)

.Organisation : Base DL/1 HIDAM/VSAM
.Lng SEGM. : mini 50, maxi 168
.Lng RECORD : 4096 octets
.DSN : &INDEX..&ROOT.&FILE.DC
.DBD : PACDCCxx
.Encombrement : suivant longueur des enregistrements

.Index primaire des éléments VisualAge Pacbase (D3)

.Organisation : Index 'DC' INDEX/VSAM
.Lng SEGM. : 29 octets
.Lng RECORD : 34 octets
.DSN : &INDEX..&ROOT.&FILE.D3
.DBD : PACDD1xx
.Encombrement : 30 enreg. par CI de 1024

.Fichier des références croisées (DX)
.Organisation : Base DL/1 HISAM/VSAM
.Lng SEGM : 80 octets
.Lng RECORD : 88 octets
.DSN : &INDEX..&ROOT.&FILE.DX
.DBD : PACDDXxx
.Encombrement : 46 enregistrements par C.I de 4096

.Fichier journal DSMS (DJ)
.Organisation : Base DL/1 HDAM/OSAM
.Lng SEGM : 187 octets
.Lng BLOCK : 4096 octets
.DSN : &INDEX..&ROOT.&FILE.DJ
.DBD : PACDDLxx
.Encombrement : 20 enreg. par block de 4096

.Fichier de travail DAF-TP (DF)
.Organisation : Base DL/1 HIDAM/VSAM
.Lng SEGM : mini 100, maxi 554
.Lng RECORD : 4096 octets
.DSN : &INDEX..&ROOT.&FILE.DF
.DBD : PACDDFxx
.Encombrement : suivant longueur enregistrement

.Index primaire du fichier de travail DAF (F1)
.Organisation : Index 'DF' INDEX/VSAM
.Lng SEGM : 37 octets
.Lng RECORD : 39 octets
.DSN : &INDEX..&ROOT.&FILE.F1
.DBD : PACDF1xx
.Encombrement : 26 enreg. par CI de 1024

.Base DSMS des S.P.A (DZ)
.Organisation : Base DL/1 HISAM/VSAM
.Lng SEGM : 8992 octets
.Lng RECORD : 9000 octets
.DSN : &INDEX..&ROOT.&FILE.DZ
.DBD : PACDDZxx
.Encombrement : 1 enreg. par CI de 10240

Trois autres fichiers séquentiels constituent la sauvegarde
du module. Ce sont :

.Fichier sauvegarde (BB)
.Organisation : Séquentielle à génération
.Recfm : Variable
.Lrecl : 354
.Blksize : 6376
.DSN : &INDEXQ..&ROOT.&ROOT.BB(n)

.Fichier archivage du journal (BJ)
.Organisation : Séquentielle à génération
.Reclsize : 180
.Codif. DSN : &INDEXQ..&ROOT.&ROOT.BJ(n)

.Fichier archive désactivée (BQ)
.Organisation : Séquentielle
.Reclsize : 180
.Codif. DSN : au choix de l'utilisateur

2.8. LA BIBLIOTHEQUE DES SOURCES POINTS D'ENTREE

LA BIBLIOTHEQUE DES COMPLEMENTS D'INSTALLATION

. Taille : 72 blocs de 6080
. Organisation : PDS
. DCB : Recfm=FB,Lrecl=80,Blksize=6080
. Dsname : &INDEX..&ROOT.&ROOT.SRC

Cette bibliothèque complémentaire contient :

-les sous-programmes de contrôle utilisateur pour les fiches améliorations, événements, sites, requêtes et maquettes

! Membre	! Contenu	!
! xxCUAM	! Contrôle TP sur fiche amélioration	!
! xxCUEV	! Contrôle TP sur fiche événement	!
! xxCUMQ	! Contrôle TP sur fiche maquette	!
! xxCURQ	! Contrôle TP sur fiche requête	!
! xxCUSI	! Contrôle TP sur fiche site	!
! PDSCAM	! Contrôle batch sur fiche amélioration	!
! PDSCEV	! Contrôle batch sur fiche événement	!
! PDSCMQ	! Contrôle batch sur fiche maquette	!
! PDSCRQ	! Contrôle batch sur fiche requête	!
! PDSCSI	! Contrôle batch sur fiche site	!

- le dictionnaire des tables DAF (DAFDIC).

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
ENVIRONNEMENT

PAGE 26

3

3. ENVIRONNEMENT

3.1. ORGANISATION DES BASES DL/1

ORGANISATION DES BASES DL/1

Le système DSMS est son propre gestionnaire de base de données. Il n'utilise DL/1 que pour stocker les enregistrements qui supportent son organisation physique.

Il en résulte que :

- . L'utilisateur ne peut en aucun cas accéder directement aux données DSMS par les utilitaires courants de DL1 mais seulement par les outils fournis dans ce but par IBM.
- . Le service demandé par DSMS à DL/1 est très réduit; en particulier, toutes les bases sont constituées du seul segment racine. Il n'y a donc pas de segment dépendant et les DBR sont de longueur fixe.
- . Il n'est jamais fait usage d'index secondaire, ni de relation logique. Il en résulte que le ratio accès physiques / accès DL/1 est sensiblement inférieur aux ratios généralement observés dans les applications courantes.

Trois types d'organisations sont utilisés: HDAM, HISAM et HIDAM.

LES BASES HDAM-OSAM

C'est l'organisation retenue pour la base journal (DJ)

La clé de cette base est une zone numérique de 7 octets, son remplissage se fait par ordre de clé croissant et sans rupture de séquence. Ce numéro d'enregistrement est un pointeur logique interne au système.

Chaque enregistrement est attaché de façon bi-univoque à un point d'ancrage (Anchor Point DL/1) calculé par le module de randomisation CGIPACR1 fourni avec le système. (Ce module est livré sous forme de source assembleur dans le fichier des paramètres 'SY' sous le nom 'RANDOM'. Il doit être compilé dans la RESLIB d'IMS du site d'implantation.) Ce module est directement déduit du module de randomisation modulo DFSHDC10 avec pour seule différence une correction du numéro de C.I. pour éviter d'adresser dans les 'BIT MAPS'.

L'attribution des emplacements dans le fichier physique est illustré par l'exemple suivant (on suppose qu'un bloc contient 25 DBR) :

Clé DSMS -----	Numéro de C.I. -----	Numéro d' A.P. -----
1	2	1
2	2	2
3	2	3
..
..
25	2	25
26	3	1
27	3	2
...
etc.....		

Il en résulte que :

- . Il n'y a jamais apparition de chaîne de synonymes, donc pas de FREE SPACE à prévoir ni d' OVERFLOW AREA.

- . L'utilisation du fichier physique se fait selon les RBA croissants, et donc l'allocation de l'espace OSAM peut se faire en fonction du volume réel occupé par la base et indépendamment du nombre de blocs adressable par le module de randomisation.

- . La description de la macro RMNAME est la suivante :
RMNAME=(CGIPACR1,X,Y) avec :

-X = nombre d' ANCHOR POINTS par bloc. Il est uniquement fonction de la taille de l'enregistrement DSMS et de la taille du bloc. Il est donné par la relation: $X = \text{ent}((BI - 19) / (ENR + 11))$ avec :

ent = fonction partie entière,

BI = taille du bloc en octets

ENR = taille de l'enregistrement DSMS en octets

-Y = nombre de blocs adressables. Compte-tenu du fait que le numéro de bloc est une fonction croissante de la clé DSMS et que cette clé est attribuée selon des numéros croissants, ce nombre peut être arbitrairement grand (sans dépasser 2 puissance 24 - 1) sans aucune influence sur l'organisation physique de la base ou sur les performances.

Il semble intéressant de lui donner dans tous les cas une valeur très élevée pour se mettre définitivement à l'abri d'un dépassement éventuel, par exemple 16 000 000.

EXEMPLE :

Considérons la base JOURNAL (DJ) :

```
.Taille de l'enregistrement DSMS           : 187
.Taille du bloc utilisé                     : 4096
----> X=20
----> RMNAME=(CGIPACR1,20,16000000)
```

LES BASES HISAM-VSAM

C'est l'organisation retenue pour les bases de documentation (DE), la base des références croisées (DX) et la base des SPA (DZ).

L'enregistrement physique DL/1 contient un et un seul DBR complet, c'est-à-dire un et un seul enregistrement DSMS plus les informations de contrôle de DL/1. Il n'y a pas de DATASET OVERFLOW. L'encombrement de la base au chargement se déduit directement du nombre d'enregistrements logiques, de la taille du RECORD, et du FREE SPACE demandé lors du DEFINE du fichier VSAM.

LES BASES HIDAM-VSAM

C'est l'organisation retenue pour la base de travail D.A.F (DF) dont l'index primaire est constitué par la base F1, et pour la base des données (DA) dont l'index primaire est constitué par la base (D1).

Chaque DBR de ces bases (DA) et (DF) ne contient qu'un segment racine de longueur variable.

Même organisation pour la base DC, éléments Visualage Pacbase, livrée avec le système Visualage Pacbase.

3.2. ENCOMBREMENT DU SYSTEME

ENCOMBREMENT DU SYSTEME

```
-----  
! les tailles sont données en pistes pour un disque 3390 !  
!-----!  
!          ! Load-modules T.P.....: 70 pistes !  
!   Les    ! Load-modules batch.....: 55 pistes !  
! Bibliothèques ! Paramètres 'SY'.....: 09 pistes !  
!   système ! DBDLIB.....: 02 pistes !  
!          ! PSBLIB.....: 05 pistes !  
!          ! !  
!-----+-----!  
! Les Fichiers ! DE0.....: !  
!   système    ! DE.....: 75 pistes !  
!          ! !  
!-----+-----!  
!          ! En pistes (3390).....:220 pistes !  
!   T O T A L ! En octets.....:12.466.080 oct.!  
!          ! ( 220 * 56.664) !  
!-----!
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
LES PROCEDURES BATCH

PAGE 31

4

4. LES PROCEDURES BATCH

4.1. PRESENTATION GENERALE

PRESENTATION GENERALE

Les traitements BATCH associés au module DSMS sont regroupés en procédures. L'objectif des chapitres suivants est de présenter chacune des procédures susceptibles d'être utilisées et d'en préciser les conditions d'exécution.

Pour chaque procédure, on trouvera :

- . Une présentation générale comprenant :
 - sa description,
 - les conditions de son 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.

4.2. CLASSIFICATION DES PROCEDURES

CLASSIFICATION DES PROCEDURES

Il existe différentes catégories de procédures batch :

LES PROCEDURES DE GESTION DE LA BASE :

- . Initialisation des fichiers DSMS (DINI),
- . Archivage des mouvements de mise à jour réalisés sur les fichiers (DARC),
- . Restauration des fichiers à partir de la sauvegarde et de l'archivage (DRST),
- . Sauvegarde des fichiers (DSAV),
- . Réorganisation du fichier de références croisées (DREO),

LES PROCEDURES UTILITAIRES :

- . Extraction du journal VA Pac des mouvements correspondant aux entités VA Pac modifiées (DEXP), liées aux améliorations,
- . Extraction du journal DSMS (DXBJ) de mouvements pour la mise à jour batch DUPT,
- . Impression des résultats de requêtes et des demandes d'édition de tables et mots-clés (DPRT),
- . Extraction de DSMS sous forme de mouvements batch d'événements, améliorations, sites ou tables (DEXT),
- . Extraction des tables pour constituer les listes de valeurs externes pour le poste développeur de la version habillée (DEXH),
- . Mise à jour batch des fichiers DSMS (DUPT,DUPD), des événements, améliorations, sites ou tables,
- . Pré-processing de sources DAF (DPDF),
- . Renommage de codes tables, sites et mots clés (DREN).
- . Impression de la liste des programmes installés (DINS).

REPRISE DE VERSIONS ANTERIEURES

Pour les sites utilisant le contrôle de VA Pac par DSMS, l'installation de DSMS 2.5 exige que VA Pac soit en 8.0.2 minimum.

LES PROCEDURES DE REPRISE DE VERSIONS ANTERIEURES :

- . Reprise de base DSMS 8.0.1 (DR80).
- . Reprise de base DSMS 8.0.2 compatible VA Pac 8.0.1 (DR8X)
A utiliser lors de la reprise VA Pac 8.0.1 en 8.0.2.
- . Reprise de base DSMS 8.0.2 01 ou 02 (DR8Q)
(reprise des requêtes).
- . Reprise de base DSMS 1.2 ou 1.5 (DR15)
- . Reprise du journal archivé DSMS 1.5 (DR5J)

REPRISE D'UNE BASE SUR UNE AUTRE PLATEFORME :

- . Remplacement des low-value par des blancs (DLVB).

4.3. 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 entrée-sortie sur les fichiers du système ou de la base provoquent une fin anormale par un ABEND USER (code 12) accompagné d'un message emis sur le fichier SYSOUT.

En présence d'un ABEND, l'utilisateur doit avant tout rechercher la présence de ce message. Il est libellé de la façon suivante :

```
**** END OF RUN DUE TO AN INPUT-OUTPUT ERROR, PROVOKED ABEND
-----
FILE : FF   OPER : OO   IKO : 0   KEY : CLE
NAME OF DATABASE           : DBDname
NAME OF SEGMENT            : SEGment name
RETURN CODE                 : code retour
PROCESSING OPTION          : Procopt
-----
APPLI aaa   NUGNA 9999   DATGN 99/99/9999   PROGR pppppp
```

Dans la plupart des cas, l'examen du code retour et le type d'opération effectuée permettent de trouver la cause de la fin anormale (ressources non disponibles, fichier trop petit, etc.).

Si ce 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 éventuellement nécessaires pour analyser le problème.

VisualAge Pacbase - Manuel d'Exploitation	PAGE	36
DSMS - EXPLOITATION & INSTALLATION		
ARCHIVAGE DU JOURNAL DES MISES A JOUR (DARC)		5

5. ARCHIVAGE DU JOURNAL DES MISES A JOUR (DARC)

5.1. PRESENTATION GENERALE

DARC : PRESENTATION GENERALE

La procédure DARC permet de sauvegarder le fichier journal (DJ) sur un fichier séquentiel (BJ) et de le réinitialiser logiquement et physiquement.

Les mises à jour archivées n'écrasent pas les archives précédentes, mais s'ajoutent à celles-ci.

Une désactivation des anciennes archives peut être demandée.

CONDITION D'EXECUTION

L'accès au TP doit être fermé.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

Si la fin anormale précède l'étape de création du fichier journal, la procédure devra être relancée telle quelle après suppression du problème.

Si la fin anormale a lieu pendant ou après l'étape de création du fichier journal, la procédure devra être relancée après modification de l'entrée utilisateur de façon à spécifier une demande de réinitialisation sans sauvegarde, le fichier journal (DJ) ayant déjà été sauvegardé.

ATTENTION :

Pour les systèmes utilisant des fichiers à génération (MVS par exemple), la version +1 du fichier archive peut avoir été cataloguée même si la procédure s'est déroulée anormalement. Dans ce cas, la procédure doit être relancée en prenant en entrée la version -1 du fichier et non la version 0.

5.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

La procédure DARC possède une entrée facultative permettant de :

- . désactiver les archives anciennes jugées obsolètes,
- . signaler l'absence en entrée d'une archive antérieure,
- . signaler la non-disponibilité en entrée du fichier des données (DA),
- . demander une réinitialisation seule du journal.

La structure de cette entrée est la suivante :

```
-----  
!Pos.! Lon.! Valeur ! Signification !  
!-----+-----+-----!  
! 2 ! 1 ! 'S' ! Code carte !  
! 3 ! 4 ! nnnn ! Numéro de session !  
! 7 ! 8 !SSAAMMJJ! OU date !  
! ! ! ! jusqu'à laquelle la désactivation !  
! ! ! ! est demandée. !  
! 15 ! 1 ! 'I' ! Absence d'archive antérieure !  
! 16 ! 1 ! 'D' ! Fichier des données (DA) indisponible!  
! 17 ! 1 ! 'J' ! Réinitialisation sans archivage !  
-----
```

Le numéro de session et la date sont exclusifs. Ils seront ignorés si l'absence d'archive antérieure est signalée.

La non-disponibilité du fichier des données n'est à signaler que lorsque ce fichier est détruit physiquement (pour plus de détails, se référer au paragraphe "Recommandations").

La demande de réinitialisation sans archivage est nécessaire lorsque le fichier journal est perdu physiquement.

ATTENTION :

Dans ce cas, l'archive précédente n'est pas recopiée sur l'archive en sortie. Si le catalogage est automatique, il existe un risque de perte des archives antérieures si on n'effectue pas de décatalogage.

En cas d'erreur sur une des options, un message d'anomalie est émis et l'archivage est exécuté avec les options par défaut.

RECOMMANDATIONS

En l'absence d'une entrée utilisateur, cette procédure ne peut être exécutée que si les données sont cohérentes et le fichier journal correctement formaté.

Lorsque les données doivent être restaurées, à la suite d'un problème, il arrive qu'une partie des informations soient détruites, ce qui empêche l'exécution de la procédure DARC, voire de la procédure DRST.

Dans ce cas de figure, et dans ce cas seulement, les colonnes 15 à 17 de l'entrée utilisateur doivent être utilisées de la façon suivante :

- . Si le fichier des données (DA) est perdu ou considéré comme étant dans un état incohérent, il convient de renseigner un D dans la colonne 16, ce qui indique au système de ne pas prendre en compte ce fichier. Il est ensuite nécessaire d'exécuter la procédure DRST car la procédure DARC, exécutée de cette façon, rend les données DA incohérentes.
- . Si le fichier journal (DJ) est perdu ou détruit, il convient de renseigner un J dans la colonne 17, ce qui permet de reformater un fichier journal vide lors de l'exécution de la procédure DARC. Il est alors possible (mais non obligatoire) d'exécuter la procédure DRST.
- . Si le fichier séquentiel archive (BJ) est perdu ou détruit, il convient de renseigner un I dans la colonne 15, ce qui implique que la procédure DARC reformate un nouveau fichier séquentiel archive.

Si par erreur une de ces colonnes est positionnée et si la procédure DARC est exécutée alors que les données DA sont dans un état cohérent, les conséquences de cette action sont les suivantes :

- . I en colonne 15 : l'archive antérieure est perdue. Tous ces mouvements peuvent être récupérés en concaténant les fichiers BJ(-1) et BJ(0) de façon à obtenir BJ(+1).
- . D en colonne 16 : la procédure DARC doit être réexécutée avant toute mise à jour. Si une mise à jour est effectuée, les données sont perdues et il faut procéder à la restauration.
- . J en colonne 17 : le contenu du fichier journal est irrémédiablement perdu.

EDITION OBTENUE

Cette procédure édite un compte rendu donnant le nombre de mises à jour archivées et éventuellement le nombre de mises à jour archivées désactivées.

RESULTAT OBTENU

Une fois la procédure effectuée, on obtient un fichier séquentiel contenant l'ensemble des mises à jour archivées.

Le journal des mises à jour effectuées en TP est réinitialisé.

Il est également possible de stocker sur un autre fichier les mises à jour qui ont été désactivées.

REMARQUE :

Cette procédure n'incrémente pas le numéro de session.

5.3. DESCRIPTION DES ETAPES

DARC : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

ARCHIVAGE DU JOURNAL : PDS300

Cette étape effectue les traitements suivants :

- . Mise à jour du fichier archive des mises à jour,
- . Positionnement d'un TOP dans le fichier des données matérialisant l'archivage du journal,
- . Ecriture des archives à désactiver sur un fichier spécifique, si la désactivation est demandée dans l'entrée utilisateur.
- . Fichiers en entrée :
 - Mouvement utilisateur
PACDMB : DSN=&&PAC7MB
 - Archive antérieure
PACDJB : DSN=&INDEXQ..&ROOT.&FILE.BJ(0)
 - Fichier journal à réinitialiser
PAC7DJ\$SUF : DSN=&INDEX..&ROOT.&FILE.DJ
 - Fichier des libellés d'erreur
PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
- . Fichier en entrée-sortie :
 - Fichier des données
PAC7DA\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7D1\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- . Fichiers en sortie :
 - Archive mise à jour
PACDBJ : DSN=&INDEXQ..&ROOT.&FILE.BJ(+1)
 - Archive désactivée
PACDBQ : DSN=DUMMY
(ce fichier peut éventuellement être récupéré)
- . Fichiers de tri
SORTWK01, SORTWK02, SORTWK03
- . Etat en sortie :
 - Compte-rendu d'archivage
PACDRU
- . Codes retour :
 - . 0 : Pas d'erreur détectée sur les fichiers.
 - . 8 : Erreur entrée utilisateur.
 - . 12 : Erreur d'entrée-sortie sur un fichier.

CREATION DU FICHIER JOURNAL : IEFBR14

Cette étape permet de recréer le journal (DJ).

- . Fichier défini :
 - Fichier journal

PAC7DJ\$SUF : DSN=&INDEX..&ROOT.&FILE.DJ

REINITIALISATION DU JOURNAL : PDS320

Cette étape effectuée 2 types de traitements :

- . Création d'un enregistrement dans le fichier journal
- . Dépositionnement du TOP du fichier des données
- . Fichiers en entrée :
 - Mouvement utilisateur
PACDMB : DSN=&&PAC7MB
 - Fichier des libellés d'erreur
PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
- . Fichier en entrée-sortie :
 - Fichier des données
PAC7DA\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- . Fichier en sortie :
 - Fichier journal à réinitialiser
PAC7DJ\$SUF : DSN=&INDEX..&ROOT.&FILE.DJ
- . Etat en sortie :
 - Compte-rendu de réinitialisation
PACDRU

5.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* - ARCHIVAGE DU JOURNAL DSMS -
//*****
//$RADP.DARC PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// VOLS='SER=$VOLO', VOLUME JOURNAL ARCHIVE
// VOLU='SER=$VOLU', VOLUME BASE 'DJ'
// INDEXP='$INDEXP', INDEX FICHER NON VSAM
// INDEXQ='$INDEXQ', INDEX FICHER GENERATION
// SPABJ='TRK,(020,10)', SPACE SAUVEGARDE JOURNAL
// SPADJ='(CYL,(3,2))', SPACE 'DJ' (OSAM DATABASE)
// UNITS='$UNITO', UNIT JOURNAL
// UNITU='$UNITU', UNIT BASE 'DJ'
// CYL=3, TAILLE WORKS DE TRI
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// SORTLIB='$BIBT', BIBLIOTHEQUE DE TRI
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* ENTREE : COMMANDE DE DESACTIVATION DE MOUVEMENTS ARCHIVES *
//* COL 2 : 'S' *
//* COL 3 A 6 : NUMERO DE SESSION *
//* COL 7 A 14 : DATE (SSAAMMJJ) *
//* COL 15 : ' ' ----> PRESENCE DU FICHER DES MVTS ARCHIVES *
//* : 'I' ----> ABSENCE DU FICHER DES MVTS ARCHIVES *
//* COL 16 : ' ' ----> PRESENCE DE LA BASE DES DONNEES (DA) *
//* : 'D' ----> ABSENCE DE LA BASE DES DONNEES (DA) *
//* COL 17 : ' ' ----> ARCHIVAGE + REINITIALISATION *
//* : 'J' ----> REINITIALISATION SANS ARCHIVAGE *
//* *
//* EN ABSENCE D'ENTREE (OU ERREUR SUR LA COMMANDE DE DESACTIVATION) *
//* AUCUNE DESACTIVATION N'A LIEU, PAR CONTRE L'ARCHIVAGE ET LA *
//* REINITIALISATION S'EXECUTENT NORMALEMENT. *
//* *
//* LES MOUVEMENTS DONT LA SESSION (LA DATE) EST ANTERIEURE OU EGALE *
//* A LA SESSION (LA DATE) INDIQUEE NE SONT PAS ARCHIVES. CEUX-CI *
//* SONT SAUVEGARDES DANS LE FICHER DES MOUVEMENTS DESACTIVES. *
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PAC7MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDS300 EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,PDS300,PDS300$SUG,&BUF,

```

```
//          &SPIE&TEST&EXCPVR&RST , &PRLD ,
//          &SRCH , &CKPTID , &MON , &LOGA , &FMTO , , , &DBRC , &IRLM)
//STEPLIB DD DSN=&RESLIB , DISP=SHR
//          DD DSN=&STEPLIB , DISP=SHR
//DFSRESLB DD DSN=&RESLIB , DISP=SHR
//IMS      DD DSN=&PSBLIB , DISP=SHR
//          DD DSN=&DBDLIB , DISP=SHR
// * : STEPCAT DD DSN=&SYSTCAT , DISP=SHR
// * :          DD DSN=&VSAMCAT , DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB , DISP=SHR
//IEFRDER  DD DUMMY ,
//          DCB=( RECFM=VB , BLKSIZE=1920 , LRECL=1916 , BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT , DCB=( RECFM=FBA , LRECL=121 ,
//          BLKSIZE=605 ) , SPACE=( 605 , ( 500 , 500 ) , RLSE , , ROUND)
//IMSUDUMP DD SYSOUT=&OUT , DCB=( RECFM=FBA , LRECL=121 ,
//          BLKSIZE=605 ) , SPACE=( 605 , ( 500 , 500 ) , RLSE , , ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP . . &ROOT . &ROOT . SY (DFSVSAM8) , DISP=SHR
//SORTLIB  DD DSN=&SORTLIB , DISP=SHR
//SORTWK01 DD UNIT=&UWK , SPACE=( CYL , &CYL , , CONTIG)
//SORTWK02 DD UNIT=&UWK , SPACE=( CYL , &CYL , , CONTIG)
//SORTWK03 DD UNIT=&UWK , SPACE=( CYL , &CYL , , CONTIG)
//PAC7DA$$SUF DD DSN=&INDEX . . &ROOT . &FILE . DA , DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX . . &ROOT . &FILE . D1 , DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX . . &ROOT . &ROOT . DE , DISP=SHR
//PAC7DJ$$SUF DD DSN=&INDEX . . &ROOT . &FILE . DJ , DISP=SHR
//PACDJB    DD DSN=&INDEXQ . . &ROOT . &FILE . BJ ( 0 ) , DISP=OLD
//PACDBJ    DD DSN=&INDEXQ . . &ROOT . &FILE . BJ ( +1 ) , DISP=( , CATLG , DELETE ) ,
//          UNIT=&UNITS , VOL=&VOLLS ,
//          SPACE=( &SPABJ , RLSE ) ,
//          DCB=&INDEXQ . . DSCB . &ROOT . &FILE . BJ
//PACDMB    DD DSN=&&PAC7MB , DISP=( OLD , PASS)
//PACDBQ    DD DUMMY , DCB=BLKSIZE=180
//PACDRU    DD SYSOUT=&OUT
// *
//OSAM1     EXEC PGM=IEHPRGM , COND=( 00 , NE , PDS300)
//DD1       DD UNIT=&UNITU , VOL=&VOLU , DISP=SHR
//SYSIN     DD DSN=&INDEXP . . &ROOT . &ROOT . SY (DL&ROOT . &FILE . DJ) , DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
// *
//OSAM2     EXEC PGM=IEFBR14 , COND=( 00 , NE , PDS300)
//PAC7DJ$$SUF DD DSN=&INDEX . . &ROOT . &FILE . DJ , UNIT=&UNITU ,
//          DISP=( , CATLG , DELETE ) , VOL=&VOLU ,
//          DCB=( RECFM=FB , LRECL=4096 , BLKSIZE=4096 ) ,
//          SPACE=&SPADJ
// *
//PDS320    EXEC PGM=DFSRR00 , REGION=$REGSIZ ,
//          PARM=( DLI , PDS320 , PDS320$$SUG , &BUF ,
//          &SPIE&TEST&EXCPVR&RST , &PRLD ,
//          &SRCH , &CKPTID , &MON , &LOGA , &FMTO , , , &DBRC , &IRLM ) ,
//          COND=( 00 , NE , PDS300)
//STEPLIB  DD DSN=&RESLIB , DISP=SHR
//          DD DSN=&STEPLIB , DISP=SHR
//DFSRESLB DD DSN=&RESLIB , DISP=SHR
//IMS      DD DSN=&PSBLIB , DISP=SHR
//          DD DSN=&DBDLIB , DISP=SHR
// * : STEPCAT DD DSN=&SYSTCAT , DISP=SHR
// * :          DD DSN=&VSAMCAT , DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB , DISP=SHR
//IEFRDER  DD DUMMY ,
//          DCB=( RECFM=VB , BLKSIZE=1920 , LRECL=1916 , BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT , DCB=( RECFM=FBA , LRECL=121 ,
//          BLKSIZE=605 ) , SPACE=( 605 , ( 500 , 500 ) , RLSE , , ROUND)
//IMSUDUMP DD SYSOUT=&OUT , DCB=( RECFM=FBA , LRECL=121 ,
//          BLKSIZE=605 ) , SPACE=( 605 , ( 500 , 500 ) , RLSE , , ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP . . &ROOT . &ROOT . SY (DFSVSAM8) , DISP=SHR
```

```
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DJ$$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,DISP=SHR
//PACDMB DD DSN=&&PAC7MB,DISP=(OLD,PASS)
//PACDRU DD SYSOUT=&OUT
//*
//DELBJ EXEC PGM=IEFBR14,COND=(08,NE,PDS300)
//*****
//DDBJ DD DSN=&INDEXQ..&ROOT.&FILE.BJ(+1),DISP=(OLD,DELETE)
```

VisualAge Pacbase - Manuel d'Exploitation	PAGE	45
DSMS - EXPLOITATION & INSTALLATION		
IMPRESSION REQUETES ET DEMANDES D'EDITION (DPRT)		6

6. IMPRESSION REQUETES ET DEMANDES D'EDITION (DPRT)

6.1. PRESENTATION GENERALE

DPRT : PRESENTATION GENERALE

La procédure DPRT assure toutes les éditions du module DSMS :

- . Les résultats des Requêtes Utilisateur sur Améliorations, Evénements et Sites, (cet ordre devant être respecté)
- . Les éditions standard de Tables, de Mots-clés, de Requêtes et de Maquettes.

La soumission -- batch et TP -- de la procédure DPRT est documentée dans le Manuel de Référence DSMS.

L'impression des éditions de Tables, de Mots-Clés de Requêtes et de maquettes ne peut être demandée qu'en batch.

Des éléments techniques sur la Fonction JOB permettant la soumission en TP de la procédure DPRT sont fournis à la fin de ce chapitre.

CONDITION D'EXECUTION

Aucune. L'accès au TP peut rester ouvert.

ANOMALIES D'EXECUTION

Se reporter au Sous-Chapitre "Anomalies d'Exécution" du Chapitre "Les Procédures Batch".

6.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

Une ligne '*' (obligatoire) :

```

+-----+-----+-----+-----+
!Pos.! Lon.! Valeur  ! Signification  !
+-----+-----+-----+-----+
!  2 !   1 !  '*'    ! Code carte      !
!  3 !   8 ! uuuuuuu ! Code utilisateur DSMS !
! 11 !   8 ! ppppppp ! Mot de passe    !
! 19 !   3 ! ppp     ! Code produit    !
! 22 !   2 ! su      ! Code filiale    !
! 24 !   1 ! 1       ! Code langue     !
+-----+-----+-----+-----+

```

Il existe 4 types d'édition possibles ; une ligne par demande est nécessaire :

```

+-----+-----+-----+-----+
!Pos.! Lon.! Valeur ! Signification  !
+-----+-----+-----+-----+
! TABLES :
+-----+-----+-----+-----+
! 02 !   03 ! Txx    ! Codes de la table Txx !
! 06 !   02 ! C1     ! ... avec leur libellé dans la langue !
!   !   !       ! de l'utilisateur connecté !
!   !   !       ! (option par défaut) !
! 06 !   02 ! C2     ! ... avec tous leurs libellés !
! 02 !   03 ! TUD    ! Codes utilisateur avec toutes leurs !
!   !   !       ! autorisations définies sur TUG, TUP !
!   !   !       ! et TUS. !
+-----+-----+-----+-----+

```

```

! REQUETES / MAQUETTES :
+-----+-----+-----+-----+
! 02 !   04 ! X QC   ! Requête sur Améliorations !
!   !   ! X QE   ! Requête sur Evénements   !
!   !   ! X QS   ! Requête sur Sites        !
! 02 !   04 ! X RC   ! Maquette sur Améliorations !
!   !   ! X RE   ! Maquette sur Evénements   !
!   !   ! X RS   ! Maquette sur Sites        !
! 06 !   06 ! xxxxxx ! Code de la requête/maquette !
! 12 !   08 ! uuuuuuu ! Utilisateur propriétaire de la !
!   !   !       ! requête ou de la maquette !
!   !   !       ! (par défaut: utilisateur connecté) !
! 20 !   02 ! C1     ! tous les écrans de descriptifs !
!   !   !       ! existants pour ce type de requête !
!   !   !       ! ou de maquette seront édités !
!   !   !       ! (option par défaut) !
!   !   ! C2     ! Seules les lignes de descriptifs !
!   !   !       ! alimentées seront éditées !
+-----+-----+-----+-----+

```

```
+-----+-----+-----+-----+
!Pos.! Lon.! Valeur ! Signification !
+-----+-----+-----+-----+
! LISTES : !
+-----+-----+-----+-----+
! 02 ! 03 ! LJQ ! Cartes de contrôle !
! 02 ! 04 ! LCQC ! Requetes sur Améliorations !
! ! ! LCQE ! Requetes sur Evénements !
! ! ! LCQS ! Requetes sur Sites !
! 02 ! 04 ! LCRC ! Maquettes sur Améliorations !
! ! ! LCRE ! Maquettes sur Evénements !
! ! ! LCRS ! Maquettes sur Sites !
! 07 ! 02 ! C1 ! Tous les écrans de descriptifs !
! ! ! ! existants pour ce type de requête !
! ! ! ! ou de maquette seront édités !
! ! ! ! (option par défaut) !
! ! ! C2 ! Seules les lignes de descriptifs !
! ! ! ! alimentées seront éditées !
! 12 ! 08 !uuuuuuuu! Utilisateur propriétaire des requêtes!
! ! ! ! ou maquettes !
+-----+-----+-----+-----+
! MOTS CLES : !
+-----+-----+-----+-----+
! 02 ! 04 ! LAKC ! Mots clés isolés des améliorations !
! ! ! LPKC ! Mots clés principaux " !
! ! ! LGKC ! Tous mots clés " !
! 06 ! 01 ! 1 ! Code langue des mots clés (langue de !
! ! ! ! (l'utilisateur connecté par défaut) !
! 02 ! 04 ! LAKE ! Mots clés isolés natifs des Evenemnts!
! ! ! LPKE ! Mots clés principaux " !
! ! ! LGKE ! Tous mots clés " !
! 02 ! 04 ! LAKT ! Mots clés isolés techn. des Evénmts. !
! ! ! LPKT ! Mots clés principaux " !
! ! ! LGKT ! Tous mots clés " !
+-----+-----+-----+-----+
```



```

+-----+
!Pos.! Lon.! Valeur ! Signification !
+-----+
!      !      !      ! .EDITION PAR REQUETE UTILISATEUR : !
!  5 !  6 ! rrrrrr ! Code de la requête utilisateur (obli-!
!      !      !      ! gatoire) - Entité "Q" utilisée. !
!  5 !  6 ! mmmmmm ! Code de la maquette (optionnel) !
! 17 !  1 ! d      ! Délimiteur (optionnel) !
!      !      !      ! Paramétrage : !
!      !      !      ! ----- !
! 18 !  1 ! s      ! Symbole - !
! 19 !  1 ! x      ! Séparateur - !
! 20 ! 54 ! ..... ! Valeurs des paramètres - !
!      !      !      ! !
!      !      !      ! Si des champs optionnels n'ont pas !
!      !      !      ! été renseignés, des valeurs par dé- !
!      !      !      ! faut sont utilisées. Elles provien- !
!      !      !      ! nent des lignes de définition de la !
!      !      !      ! requête de l'utilisateur trouvées !
!      !      !      ! dans la base de données. !
+-----+

```

EDITION OBTENUE

Deux types d'édition :

- . Les résultats des Requetes de l'utilisateur sur Evénements, Améliorations ou Sites.

- . Les éditions standard des Tables, des Mots-clés, des Requetes et des Maquettes.

CODE RETOUR

```

+-----+
!  0 ! OK avec requêtes !
!  4 ! OK avec demandes d'édition tables, mc, Req, maq. !
!  8 ! OK mais des requêtes ou demandes sont erronées !
! 12 ! Erreur fatale !
! 16 ! Erreur dans le tri !
+-----+

```

6.3. DESCRIPTION DES ETAPES

DPRT : DESCRIPTION DES ETAPES

Cette procédure fait appel à un programme unique (PDSB) qui sert de moniteur d'enchaînement des différents programmes, considérés comme sous-programmes de ce moniteur.

Elle comprend les étapes suivantes :

PRISE EN COMPTE DES ENTREES : PTU001

Les entrées sont automatiquement mises en forme lorsque les REQUETES sont soumises en TP.

VERIFICATION DES FICHIERS VSAM : IDCAMS

EDITIONS : PDSB

- . Fichiers permanents en entrée :
 - Fichier des données
 - PAC7DA\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7D1\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
 - Fichier des éléments VisualAge Pacbase
 - PAC7DC\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
 - PAC7D3\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
 - Fichier des libellés d'erreur
 - PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
- . Fichier en entrée :
 - Requetes utilisateur
 - PACDMB : DSN=&&PACDMB
- . Fichiers de travail :
 - Demandes d'édition
 - PACDKD
 - Requetes
 - PACDKQ
 - Temporaires
 - PACDW1, PACDW2, PACDW3, PACDW4
- Fichiers de tri
 - SORTWK01, SORTWK02, SORTWK03
- . Etats en sortie :
 - Compte-rendu d'enchaînement
 - PACDIA
 - Liste des requêtes et demandes
 - PACDIB
 - Edition des tables et mots-clés
 - PACDID
 - Compte-rendu d'extractions par requête
 - PACDIQ
 - Edition des résultats d'extraction
 - PACDQI
 - Edition des requêtes/maquettes
 - PACDRQ
 - Edition des cartes de contrôle
 - PACDJQ

6.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5 *
//* - EDITIONS ET REQUETES - *
//*****
//$RADP.DPRT PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// OUTL='$OUT', CLASSE DE SORTIE DES ETATS
// SPAMB='TRK,(30,05)', SPACE COMMANDES EXTRACTION
// SPAWK='CYL,(20,02)', SPACE FICHIERS DE TRAVAIL
// UWK=$UWK, UNITE DE TRAVAIL
// CYL=3, TAILLE WORKS DE TRI
// COPIES=1, NOMBRE D'EXEMPLAIRES DES ETATS
// LNG=, LANGAGE MONITEUR (ANGLAIS = 'E')
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// SORTLIB='$BIBT', BIBLIOTHEQUE DE TRI
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// DBDLIO='$DBDLIO', BIBLIOTHEQUE DES DBD VA PAC
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPAMB),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSB EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSB&LNG,PDSB$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)

```

```
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,  
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)  
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,  
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)  
//IMSMON DD DUMMY  
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR  
//SORTLIB DD DSN=&SORTLIB,DISP=SHR  
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR  
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR  
//PAC7DC$$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR  
//PAC7D3$$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR  
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR  
//PACDIA DD SYSOUT=&OUTL  
//PACDIB DD SYSOUT=&OUTL  
//PACDID DD SYSOUT=&OUTL,COPIES=&COPIES  
//PACDIQ DD SYSOUT=&OUTL,COPIES=&COPIES  
//PACDQI DD SYSOUT=&OUTL,COPIES=&COPIES  
//PACDRQ DD SYSOUT=&OUTL,COPIES=&COPIES  
//PACDJQ DD SYSOUT=&OUTL,COPIES=&COPIES  
//PACDKD DD UNIT=&UWK,SPACE=(&SPAMB),DCB=BLKSIZE=6256  
//PACDKQ DD UNIT=&UWK,SPACE=(&SPAMB),DCB=BLKSIZE=6160  
//PACDMB DD DSN=&&PACDMB,DISP=(OLD,DELETE,DELETE)  
//PACDW1 DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6160  
//PACDW2 DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6080  
//PACDW3 DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6375  
//PACDW4 DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6080  
//*
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
RESTAURATION

(DRST)

PAGE 53

7

7. RESTAURATION

(DRST)

7.1. PRESENTATION GENERALE

DRST : PRESENTATION GENERALE

Cette procédure a pour fonction la restauration des fichiers à partir de l'image séquentielle obtenue par la procédure de sauvegarde (DSAV).

Elle permet également de récupérer les mises à jour archivées après obtention de cette sauvegarde.

CONDITION D'EXECUTION

Les fichiers doivent être fermés au TP.

Ces modifications doivent se faire dans la bibliothèque des paramètres d'exploitation SY.

La procédure réinitialise physiquement et logiquement le journal ; celui-ci doit donc avoir été préalablement sauvegardé par la procédure d'archivage (DARC).

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre 'ANOMALIES D'EXECUTION' du chapitre "LES PROCEDURES BATCH".

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

REMARQUE

Cette procédure recréant les fichiers, il peut être intéressant de réajuster leur taille en fonction de leur évolution estimée.

SOUS-PROGRAMMES DE CONTROLE DES DEFINITIONS

Des sous-programmes (livrés sous forme de source COBOL) permettent d'ajouter des contrôles spécifiques ou des initialisations sur les 5 définitions de DSMS.

Ces sources ne comportent au départ que 3 exemples :

- 1 erreur de type 'WARNING',
- 1 erreur sévère,
- 1 initialisation.

Leur linkage est constituée des zones affichées, des zones saisies et de quelques autres associées directement ou non à la définition concernée.

En retour de ces sous-programmes, il est donc possible de faire afficher un message d'erreur ou bien d'écraser les valeurs des zones affichées.

REMARQUES : .Les contrôles habituels des définitions sont effectués avant et après leur appel.

.Les positionnements d'erreurs de type "WARNING" provoquent l'envoi d'un message sur l'écran de Définition suivi d'un rappel au sous-programme pour que celui-ci réinitialise le PR positionné à 'W'.

L'appel à ces sous-programmes est déclenché par des tops renseignés dans l'enregistrement technique de la procédure DRST.

7.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

La structure de l'entrée est la suivante :

```

+-----+-----+-----+-----+
!Pos.! Lon.! Valeur ! Signification !
+-----+-----+-----+-----+
! 2 ! 1 ! 'R' ! Code carte !
! 3 ! 1 ! 'l' ! Code langue 'E' ou 'F' (facultatif) !
! 4 ! 1 ! ! Top inhibition du journal !
! ! ! '0' ! Pas d'inhibition (Option par défaut) !
! ! ! '1' ! Inhibition de la journalisation !
! 5 ! 1 ! ! Inutilisé !
! 6 ! 3 ! 'REC' ! Restauration avec récupération des !
! ! ! ! mises à jour archivées !
! 9 ! 12 ! ! Table de 12 postes permettant d'in- !
! ! ! ! diquer la signification des touches !
! ! ! ! fonction !
! ! ! ! (par défaut: 123456789ABC, il est !
! ! ! ! possible de déplacer ou de remettre !
! ! ! ! à blanc une ou plusieurs valeurs) !
! 21 ! 1 ! ! INTERFACE SYSTEMES DE SECURITE !
! ! ! ' ' ! Reprise de la valeur précédente !
! ! ! ! ou pas d'interface (en création) !
! ! ! '&' ! Remise à blanc = Désactivation !
! ! ! 'R' ! RACF !
! ! ! 'S' ! TOPSECRET !
! 22 ! 1 ! ! CONTROLE UTILISATEUR SOUS RACF EN TP !
! ! ! ' ' ! Reprise de la valeur précédente !
! ! ! '&' ! Remise à blanc = possibilité d'entrer !
! ! ! ! un autre utilisateur-mot de passe !
! ! ! ! que celui de la connexion initiale !
! ! ! 'N' ! Pas de possibilité d'entrer un autre !
! ! ! ! utilisateur-mot de passe !
! 23 ! 1 ! 'C' ! Cryptage des mots de passe !
! ! ! 'D' ! Décryptage des mots de passe !
! ! ! ' ' ! Mots de passe inchangés !
! ! ! ! REMARQUE: il est fortement déconseil- !
! ! ! ! lé de demander un cryptage ou dé- !
! ! ! ! cryptage des mots de passe en même !
! ! ! ! temps que la récupération des mou- !
! ! ! ! vements archivés (l'action n'étant !
! ! ! ! pas effectuée sur le journal) !
+-----+-----+-----+-----+

```



```

+-----+-----+-----+-----+
!Pos.! Lon.! Valeur ! Signification !
+-----+-----+-----+-----+
! 26 ! 1 ! 'C' ! Appel du sous-pgm de contrôles sup- !
! ! ! ! plémentaires pour fiche amélioration !
! ! ! ! ' & ' ! Pas d'appel du sous-programme !
! 27 ! 1 ! 'E' ! Appel du sous-pgm de contrôles sup- !
! ! ! ! plémentaires pour fiche événement !
! ! ! ! ' & ' ! Pas d'appel du sous-programme !
! 28 ! 1 ! 'Q' ! Appel du sous-pgm de contrôles sup- !
! ! ! ! plémentaires pour fiche requête !
! ! ! ! ' & ' ! Pas d'appel du sous-programme !
! 29 ! 1 ! 'R' ! Appel du sous-pgm de contrôles sup- !
! ! ! ! plémentaires pour fiche maquette !
! ! ! ! ' & ' ! Pas d'appel du sous-programme !
! 30 ! 1 ! 'S' ! Appel du sous-pgm de contrôles sup- !
! ! ! ! plémentaires pour fiche site !
! ! ! ! ' & ' ! Pas d'appel du sous-programme !
+-----+-----+-----+-----+

```

EDITION OBTENUE

Cette procédure édite un compte rendu donnant les options demandées, les erreurs éventuellement associées, le nombre d'enregistrements rechargés pour chacun des fichiers, les options mémorisées.

RESULTAT OBTENU

Une fois la procédure exécutée, le numéro de session courant est celui de l'image séquentielle, ou celui de la mise à jour la plus récente si la récupération des mises à jour archivées a été demandée.

7.3. DESCRIPTION DES ETAPES

DRST : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

CONTROLE EXISTENCE DU JOURNAL : IDCAMS

Cette étape effectue un LISTCAT sur le fichier journal (DJ).
Elle renvoie comme code retour :

.0 : Le fichier journal existe
.AUTRE : Il n'existe pas

CONTROLE DU CONTENU DU JOURNAL : PDS380

Cette étape n'est exécutée que si le fichier journal existe.
Dans ce cas, elle va vérifier s'il a été archivé.

. Fichiers en entrée :

-Fichier journal
PAC7DJ\$\$SUF: DSN=&INDEX..&ROOT.\$FILE.DJ
-Fichier des libellés d'erreur
PAC7DE\$\$SUF: DSN=&INDEX..&ROOT.&ROOT.DE

. Etat en sortie

- Etat du fichier AJ
PACDRU

Il est édité si le fichier journal n'a pas été archivé.

. Codes retour :

-0 : Le fichier journal a été archivé.
-4 : Le fichier journal n'a pas été archivé.
(Aucune étape de DRST n'est exécutée).

DEFINITION DES FICHIERS : IDCAMS

Cette étape est exécutée si le fichier journal a été archivé. Elle contient les DELETE/DEFINE des fichiers de la base.

DEFINITION DES FICHIERS : IEFBR14

Cette étape exécute un DELETE/DEFINE du fichier JOURNAL.

RESTAURATION DE LA BASE DSMS : PDS400

Cette étape n'est exécutée que si le fichier journal a été archivé.

```
. Fichiers permanents en entrée :
- Sauvegarde des fichiers
PACDBB:          DSN=&INDEXQ.&ROOT.&FILE.BB(0)
-Fichier des libellés d'erreur
PAC7DE$$SUF:    DSN=&INDEX.&ROOT.&ROOT.DE

. Fichiers permanents en sortie :
-Fichier des données
PAC7DA$$SUF:    DSN=&INDEX.&ROOT.&FILE.DA
PAC7D1$$SUF:    DSN=&INDEX.&ROOT.&FILE.D1
-Fichier des éléments VA Pac
PAC7DC$$SUF:    DSN=&INDEX.&ROOT.&FILE.DC
PAC7D3$$SUF:    DSN=&INDEX.&ROOT.&FILE.D3
-Fichier journal
PAC7DJ$$SUF:    DSN=&INDEX.&ROOT.&FILE.DJ
-Fichier des références croisées
PAC7DX$$SUF:    DSN=&INDEX.&ROOT.&FILE.DX

. Fichier mouvement en entrée :
- Mouvements utilisateurs
PACDMB :        DSN=&&RESTMB

. Fichier en sortie :
- Fichier de travail (2 enreg.)
PACDMS :        DSN=&&PACDMS

. Etat en sortie
- Compte-rendu de restauration
PACDRU
```

REAPPLICATION DE L'ARCHIVE : PDS450

Cette étape n'est exécutée que s'il y a des mouvements à récupérer. Elle ne provoque pas de journalisation des mouvements passés.

```
. Fichiers permanents en entrée-sortie :
-Fichier des données
PAC7DA$$SUF:    DSN=&INDEX.&ROOT.&FILE.DA
PAC7D1$$SUF:    DSN=&INDEX.&ROOT.&FILE.D1
-Fichier des éléments VisualAge Pacbase
PAC7DC$$SUF:    DSN=&INDEX.&ROOT.&FILE.DC
PAC7D3$$SUF:    DSN=&INDEX.&ROOT.&FILE.D3
-Fichier des références croisées
PAC7DX$$SUF:    DSN=&INDEX.&ROOT.&FILE.DX

. Fichiers en entrée :
-Fichier de travail (2 enreg.)
PACDMS :        DSN=&&PACDMS
-Fichier des libellés d'erreur
```

RESTAURATION
DESCRIPTION DES ETAPES

(DRST)

PAGE

60

7
3

PAC7DE\$SUF: DSN=&INDEX..&ROOT.&ROOT.DE

. Fichier archive en entrée :
-Archivage journal à réappliquer
PACDBJ : DSN=&INDEXQ..&ROOT.&FILE.BJ(0)

. Etat en sortie :
-Compte-rendu de mise à jour
PACDRU

7.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* - CHARGEMENT-RESTAURATION BASE DSMS -
//*****
//$RADP.DRST PROC FILE=$FILE, NO DE BASE PHYSIQUE
// FILEO=$FILEO, NO DE BASE PHYSIQUE VA PAC
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// ROOTO=$ROOTO, RADICAL DU SYSTEME VA PAC
// INDEX='$INDEX', INDEX VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// INDEXO='$INDEXO', INDEX FICHIERS NON VSAM VA PAC
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
// INDEXQ='$INDEXQ', INDEX FICHIERS A GENERATIONS
// SPADJ='(CYL,(3,2))', SPACE 'DJ' (BASE OSAM)
// VOLU='$VOLU', VOLUME BASE 'DJ'
// UNITU='$UNITU', UNIT BASE 'DJ'
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// DBDLIO='$DBDLIO', BIBLIOTHEQUE DES DBD VA PAC
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* ENTREE
//* COL 02 : 'R'
//* COL 03 : CODE LANGUE INITIAL (F=FRANCAIS, E=ANGLAIS)
//* COL 04 : '1' --> INHIBITION DU LOG DES MOUVEMENTS
//* COL 05 : FORMAT DATE MACHINE (N --> POUR JJ/MM/AA)
//* : (I --> POUR MM/JJ/AA)
//* COL 06-08 : 'REC' POUR RECUPERATION DES MOUVEMENTS ARCHIVES
//* : 'INI' POUR INITIALISATION DES FICHIERS
//* COL 09-20 : SIGNIFICATION DES TOUCHES FONCTION
//* COL 21 : INTERFACE SYSTEME DE SECURITE
//* : ' ' --> REPRISE VALEUR PRECEDENTE OU PAS INTERFACE
//* : '&' --> REMISE A BLANC (DESACTIVATION)
//* : 'R' --> RACF
//* COL 22 : CONTROLE UTILISATEUR SOUS RACF EN TP
//* : ' ' --> REPRISE VALEUR PRECEDENTE
//* : '&' --> REMISE A BLANC (POSSIBILITE D'ENTRER UN
//* : AUTRE UTILISATEUR-MOT DE PASSE QUE CELUI
//* : DE LA CONNEXION)
//* : 'N' --> PAS DE POSSIBILITE D'ENTRER UN AUTRE
//* : UTILISATEUR-MOT DE PASSE
//* COL 23 : 'C' --> CRYPTAGE DU MOT DE PASSE
//* : 'D' --> DECRYPTAGE DU MOT DE PASSE
//* : ' ' --> MOTS DE PASSE INCHANGES
//* COL 26 : 'C' --> APPEL DU SOUS-PGM DE CONTROLES SUPPLEMEN-
//* : TAIRES POUR FICHE AMELIORATION
//* : '&' --> PAS D'APPEL DU SOUS-PGM
//* COL 27 : 'E' --> APPEL DU SOUS-PGM DE CONTROLES SUPPLEMEN-
//* : TAIRES POUR FICHE EVENEMENT
//* : '&' --> PAS D'APPEL DU SOUS-PGM
//* COL 28 : 'Q' --> APPEL DU SOUS-PGM DE CONTROLES SUPPLEMEN-
//* : TAIRES POUR FICHE REQUETE
//* : '&' --> PAS D'APPEL DU SOUS-PGM
//* COL 29 : 'R' --> APPEL DU SOUS-PGM DE CONTROLES SUPPLEMEN-
//* : TAIRES POUR FICHE MAQUETTE
//* : '&' --> PAS D'APPEL DU SOUS-PGM
//* COL 30 : 'S' --> APPEL DU SOUS-PGM DE CONTROLES SUPPLEMEN-
//* : TAIRES POUR FICHE SITE
//* : '&' --> PAS D'APPEL DU SOUS-PGM
//*
//* SI LA BASE JOURNAL DES MOUVEMENTS SUR DISQUE (DJ) N'EST PAS

```

```

// * REINITIALISEE, LA CHAINE DE RESTAURATION N'EST PAS EXECUTEE; *
// * IL FAUT ALORS AU PREALABLE EXECUTER LA PROCEDURE 'DARC'. *
// *****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&DRSTMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
// *
//EXISDJ EXEC PGM=IDCAMS
// *:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(LI&ROOT.&FILE.DJ),DISP=SHR
// *
//PDS380 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDS380,PDS380$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
// COND=(00,NE,EXISDJ)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// *:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// *: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DJ$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,DISP=SHR
//PACDRU DD SYSOUT=&OUT
// *
//DEFINE EXEC PGM=IDCAMS,COND=(00,NE,PDS380)
// *:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.DA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.D1),DISP=SHR
// DD DSN=&INDEXO..&ROOTO.&ROOTO.SY(DF&ROOTO.&FILEO.DC),DISP=SHR
// DD DSN=&INDEXO..&ROOTO.&ROOTO.SY(DF&ROOTO.&FILEO.D3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.DX),DISP=SHR
// *
//OSAMDJ1 EXEC PGM=IEHPRGM,COND=(00,NE,PDS380)
//DD1 DD UNIT=&UNITU,VOL=SER=&VOLU,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(DL&ROOT.&FILE.DJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
// *
//OSAMDJ2 EXEC PGM=IEFBR14,COND=(00,NE,PDS380)
//PAC7DJ$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,UNIT=&UNITU,
// DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
// DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
// SPACE=&SPADJ
// *
//PDS400 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDS400,PDS400$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
// COND=(00,NE,PDS380)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR

```

```
//          DD DSN=&DBDLIO,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DJ$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,DISP=SHR
//PAC7DX$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PACDBB   DD DSN=&INDEXQ..&ROOT.&FILE.BB(0),DISP=OLD
//PACDMB   DD DSN=&&DRSTMB,DISP=(OLD,PASS)
//PACDMS   DD DSN=&&PACDMS,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//          SPACE=(TRK,(1,1))
//PACDRU   DD SYSOUT=&OUT
//*
//PDS450   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,PDS450,PDS450$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
//          COND=((00,NE,PDS380),(00,NE,PDS400))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//          DD DSN=&DBDLIO,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DX$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PACDBJ   DD DSN=&INDEXQ..&ROOT.&FILE.BJ(0),DISP=SHR
//PACDMS   DD DSN=&&PACDMS,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//*
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
SAUVEGARDE

	PAGE	
	64	
(DSAV)		8

8. SAUVEGARDE

(DSAV)

8.1. PRESENTATION GENERALE

SAUVEGARDE (DSAV) : PRESENTATION GENERALE

La procédure de sauvegarde DSAV a pour but de mettre l'ensemble des fichiers principaux constituant le module DSMS sous forme séquentielle BB.

Les fichiers qui sont sauvegardés sont les suivants :

- . Le fichier des données (DA),
- . Le fichier des éléments VA Pac (DC),
- . Le fichier des références croisées (DX).

CONDITION D'EXECUTION

L'accès au TP doit être fermé, afin de conserver la cohérence des fichiers pendant la sauvegarde.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures batch".

La cause principale d'une fin anormale est l'oubli de fermeture du TP.

Après correction, la procédure peut être relancée telle quelle dans tous les cas.

8.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

Une carte facultative :

```
-----  
!Pos.! Lon.! Valeur ! Signification !  
!-----+-----+-----!  
! 2 ! 1 ! 'O' ! Code carte !  
! 3 ! 3 ! 'ENC' ! Cryptage des mots de passe !  
! ! ! 'DEC' ! Décryptage des mots de passe !  
! ! ! ' ' ! Mots de passe inchangés !  
-----
```

EDITION OBTENUE

Une fois la sauvegarde effectuée, cette procédure édite un compte-rendu donnant le nombre d'enregistrements sauvegardés pour chacun des fichiers et le numéro de session.

RESULTAT OBTENU

Le résultat obtenu est un fichier séquentiel unique (BB), de longueur variable, contenant l'image des 3 fichiers sauvegardés.

Si la base est dans un état incohérent à la suite d'une "fin anormale de la dernière mise à jour", la sauvegarde ne sera pas exécutée.

REMARQUE :

Cette procédure incrémente le numéro de session.

8.3. DESCRIPTION DES ETAPES

DSAV : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

VERIFICATION INTEGRITE DE LA BASE : PDSBAS

. Fichiers permanents en entrée :
- Fichier des données
 PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- Fichier des libellés d'erreur
 PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Etat en sortie :
- Compte rendu de validité
 PACDRS

. Code retour :

Cet utilitaire envoie un code retour 4 et provoque un ABEND en cas d'invalidité des données.

SAUVEGARDE DE LA BASE : PDS500

. Fichier en entrée-sortie :
- Fichier des données
 PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

. Fichiers permanents en entrée :
- Fichier des éléments VisualAge Pacbase
 PAC7DC\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
 PAC7D3\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
- Fichier des références croisées
 PAC7DX\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DX
- Fichier des libellés d'erreur
 PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Fichier mouvement en entrée :
- Mouvements utilisateurs
 PACDMB : DSN=&&DSAVMB

. Fichier en sortie :
- Image séquentielle des fichiers
 PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

. Etat en sortie :
- Compte rendu de sauvegarde
 PACDRU

8.4. JCL DE LA PROCEDURE

```
//*****  
//* DSMS 2.5 *  
//* - SAUVEGARDE DE LA BASE DSMS - *  
//*****  
//$RADP.DSAV PROC FILE=$FILE, NO DE BASE PHYSIQUE  
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS  
// INDEX='$INDEX', INDEX VSAM  
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME  
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR  
// OUT='$OUT', CLASSE DE SORTIE  
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM  
// INDEXQ='$INDEXQ', INDEX FICHIERS A GENERATIONS  
// VOLS='SER=$VOLO', VOLUME FICHER GENERATION  
// UNITS='$UNITO', UNITE FICHER GENERATION  
// SPABB='TRK,(10,2)', SPACE SAUVEGARDE  
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH  
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB  
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD  
// DBDLIO='$DBDLIO', BIBLIOTHEQUE DES DBD VA PAC  
// RESLIB='$RESLIB', RESLIB IMS  
// PROCLIB='$PROCLIB', PROCLIB IMS  
// UWK='$UWK', UNITE DE TRAVAIL  
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,  
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM  
//*-----*  
//INPUT EXEC PGM=PTU001  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80  
//PAC7MB DD DSN=&&DSAVMB,DISP=(,PASS),  
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)  
//*  
//VERIFY EXEC PGM=IDCAMS  
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR  
//*: DD DSN=&VSAMCAT,DISP=SHR  
//SYSPRINT DD SYSOUT=&OUT  
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR  
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR  
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR  
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR  
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR  
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR  
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR  
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR  
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR  
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR  
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR  
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR  
//*  
//PDSBAS EXEC PGM=DFSRR00,REGION=$REGSIZ,  
// PARM=(DLI,PDSBAS,PDSBAS$SUG,&BUF,  
// &SPIE&TEST&EXCPVR&RST,&PRLD,  
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM)  
//STEPLIB DD DSN=&RESLIB,DISP=SHR  
// DD DSN=&STEPLIB,DISP=SHR  
//DFSRESLB DD DSN=&RESLIB,DISP=SHR  
//IMS DD DSN=&PSBLIB,DISP=SHR  
// DD DSN=&DBDLIB,DISP=SHR  
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR  
//*: DD DSN=&VSAMCAT,DISP=SHR  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//DDSNAP DD SYSOUT=&OUT  
//PROCLIB DD DSN=&PROCLIB,DISP=SHR  
//IEFRDER DD DUMMY,  
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)  
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,  
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
```

```
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDRS    DD SYSOUT=&OUT
//*
//PDS500   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,PDS500,PDS500$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
//          COND=(00,NE,PDSBAS)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//          DD DSN=&DBDLIO,DISP=SHR
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC$$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3$$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DX$$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PACDMB    DD DSN=&&DSAVMB,DISP=(OLD,PASS)
//PACDBB    DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,
//          SPACE=(&SPABB,RLSE),
//          DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB
//PACDRU    DD SYSOUT=&OUT
//*
//DELB    EXEC PGM=IEFBR14,COND=(08,NE,PDS500)
//*****
//DDBB    DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(OLD,DELETE)
```

VisualAge Pacbase - Manuel d'Exploitation	PAGE	70
DSMS - EXPLOITATION & INSTALLATION		
REORGANISATION DES REFERENCES CROISEES (DREO)		9

9. REORGANISATION DES REFERENCES CROISEES (DREO)

REORGANISATION DES REFERENCES CROISEES	(DREO)	PAGE	71
PRESENTATION GENERALE			9 1

9.1. PRESENTATION GENERALE

REORGANISATION DES REFERENCES CROISEES : INTRODUCTION

La procédure de réorganisation (DREO) a pour but de reconstituer une image séquentielle de la base à partir d'une autre image séquentielle. Le fichier obtenu servira d'entrée à la procédure de restauration (DRST).

Le principe de fonctionnement de cette procédure est de reconstituer les références croisées associées aux données à partir de l'image de ces données.

CONDITION D'EXECUTION

La base, si elle est disponible, peut rester ouverte pendant la réorganisation, puisque celle-ci fonctionne sur des images séquentielles de la base (sauvegardes).

Les mises à jour effectuées postérieurement à la constitution de la sauvegarde qui a servi à la réorganisation seront récupérables lors de la restauration de la base réorganisée.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

En cas de fin anormale d'une des étapes, la réorganisation doit être reprise au début.

9.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

Trois différents types d'entrées utilisateurs peuvent être utilisés, mais il n'est possible de créer qu'une seule ligne de chaque type.

La structure de l'entrée est la suivante :

```

-----
!Pos.! Lon.! Valeur  !Signification
!-----+-----+-----+-----!
!  1 !  1 !Inutilisé!
!   !   !   !
!   !   !   !
!  2 !  1 ! 'P'    !Destruction de Produits
!   !   ! 'S'    !Destruction de Filiales
!   !   ! 'X'    !Destruction de Produits/Filiales
!   !   !   !
!   !   !   !
!  3 ! 60 ! Code  !(20 x 3 car.) si col.2 = 'P'
!   !   ! Produit !
!   !   !   !
!   ! 60 ! Code  !(30 x 2 car.) si col.2 = 'S'
!   !   ! Filiale !
!   !   !   !
!   ! 60 ! Produit/!(12 x 5 car.) si col.2 = 'X'
!   !   ! Filiale !
!   !   !   !
!-----+-----+-----+-----!

```

EDITION OBTENUE

Cette procédure édite des messages sur les incohérences éventuellement rencontrées dans le fichier des données.

RESULTAT OBTENU

Le résultat de cette procédure est une image séquentielle réorganisée de la base DSMS, sauvegarde constituant l'entrée de la procédure de restauration (DRST).

9.3. DESCRIPTION DES ETAPES

DREO : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

CONSTITUTION INDEX (non mots-clés) : PDSR10

- . Fichier en entrée :
 - Mouvements
CARTE : DSN=&&PACDMB
- . Fichiers permanents en entrée :
 - Sauvegarde de la base DSMS
PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(0)
 - Fichier des libellés d'erreur
PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
- . Fichiers de travail
 - Données et éléments VA Pac
PACDW1 : DSN=&&W1
 - Mots-clés et Références mots-clés
PACDW2 : DSN=&&W2
 - Références croisées non mots-clés
PACDW3 : DSN=&&W3
 - Fichiers de tri
SORTWK01, 02, 03
- . Etats en sortie :
 - Incohérences sur les données DSMS
PACDRH
 - Compte-rendu de réorganisation
PACDRK

CONSTITUTION INDEX MOTS-CLES : PDSR20

- . Fichiers de travail
 - Mots-clés et Références mots-clés
PACDW2 : DSN=&&W2
 - Mots-clés
PACDW4 : DSN=&&W4
 - Références mots-clés
PACDW5 : DSN=&&W5
 - Fichiers de tri
SORTWK01, 02, 03

FUSION DES INDEX : PDSR30

- . Fichiers de travail
 - Références croisées non mots-clés
PACDW3 : DSN=&&W3
 - Références mots-clés
PACDW5 : DSN=&&W5
 - Références mots-clés
PACDW6 : DSN=&&W6
 - Fichiers de tri
SORTWK01, 02, 03

FUSION GENERALE POUR SAUVEGARDE : PDSR40

- . Fichiers de travail
 - Données et éléments VA Pac
PACDW1 : DSN=&&W1
 - Mots-clés

PACDW4 : DSN=&&W4
- Références mots-clés
PACDW6 : DSN=&&W6
- Fichiers de tri
SORTWK01, 02, 03

. Fichier permanent en entrée :
- Fichier des libellés d'erreur
PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Fichier permanent en sortie :
- Sauvegarde base DSMS réorganisée
PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

. Etat en sortie :
- Compte-rendu de réorganisation
PACDRR

9.4. JCL DE LA PROCEDURE

```

//*****
// * DSMS 2.5 *
// * - REORGANISATION DSMS - *
//*****
//$RADP.DREO PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// OUT=' $OUT', CLASSE DE SORTIE
// INDEX=' $INDEX', INDEX VSAM
// INDEXP=' $INDEXP', INDEX FICHIERS NON VSAM
// INDEXQ=' $INDEXQ', INDEX FICHIERS A GENERATIONS
//*: SYSTCAT=' $CATV', VSAM SYSTEM CATALOG
// CYL=' (4,1)', TAILLE WORKS DE TRI
// SPADA='TRK,(60,5)', SPACE DE TRAVAIL (DA + DC)
// SPADX='TRK,(60,5)', SPACE DE TRAVAIL (DX)
// SPABB='TRK,(20,5)', SPACE SAUVEGARDE
// VOLS='SER=$VOLO', VOLUME FICHER GENERATION
// UNITS=' $UNITO', UNITE FICHER GENERATION
// STEPLIB=' $MODB', BIBLIOTHEQUE BATCH
// SORTLIB=' $BIBT', BIBLIOTHEQUE DE TRI
// UWK=$UWK, UNITE DE TRAVAIL
// PSBLIB=' $PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB=' $DBDLIB', BIBLIOTHEQUE DES DBD
// RESLIB=' $RESLIB', RESLIB IMS
// PROCLIB=' $PRCLIB', PROCLIB IMS
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//-----*
// *
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
// *
//PDSR10 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSR10,PDSR10$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP.&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DE$SUF DD DSN=&INDEX.&ROOT.&ROOT.DE,DISP=SHR
//CARTE DD DSN=&&PACDMB,DISP=(OLD,DELETE)
//PACDBB DD DSN=$INDEXQ.&ROOT.&FILE.BB(0),DISP=SHR
//PACDW1 DD DSN=&&W1,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPADA,RLSE),
// DCB=(RECFM=VB,LRECL=354,BLKSIZE=6022)
//PACDW2 DD DSN=&&W2,DISP=(,PASS),

```

```
//          UNIT=&UWK,SPACE=( &SPADX,RLSE),
//          DCB=(RECFM=FB,LRECL=120,BLKSIZE=6240)
//PACDW3 DD DSN=&W3,DISP=(,PASS),
//          UNIT=&UWK,SPACE=( &SPADA,RLSE),
//          DCB=(RECFM=FB,LRECL=080,BLKSIZE=6400)
//PACDRH DD SYSOUT=&OUT
//PACDRK DD SYSOUT=&OUT
//*
//PDSR20 EXEC PGM=PDSR20
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDW2 DD DSN=&W2,DISP=(OLD,PASS)
//PACDW4 DD DSN=&W4,DISP=(,PASS),
//          UNIT=&UWK,SPACE=( &SPADX,RLSE),
//          DCB=(RECFM=FB,LRECL=340,BLKSIZE=6120)
//PACDW5 DD DSN=&W5,DISP=(,PASS),
//          UNIT=&UWK,SPACE=( &SPADA,RLSE),
//          DCB=(RECFM=FB,LRECL=080,BLKSIZE=6400)
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PDSR30 EXEC PGM=PDSR30
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDW3 DD DSN=&W3,DISP=(OLD,PASS)
//PACDW5 DD DSN=&W5,DISP=(OLD,PASS)
//PACDW6 DD DSN=&W6,DISP=(,PASS),
//          UNIT=&UWK,SPACE=( &SPADA,RLSE),
//          DCB=(RECFM=FB,LRECL=080,BLKSIZE=6400)
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PDSR40 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,PDSR40,PDSR40$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDBB DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOL,S,
//          SPACE=( &SPABB,RLSE),
//          DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB
//PACDW1 DD DSN=&W1,DISP=(OLD,PASS)
//PACDW4 DD DSN=&W4,DISP=(OLD,PASS)
```

```
//PACDW6 DD DSN=&&W6,DISP=(OLD,PASS)
//PACDRR DD SYSOUT=&OUT
//*
//DEL12 EXEC PGM=IEFBR14,COND=(12,NE,PDSR40)
//*****
//DDBB DD DSN=&INDEXQ.&ROOT.&FILE.BB(+1),DISP=(OLD,DELETE)
```

VisualAge Pacbase - Manuel d'Exploitation	PAGE	78
DSMS - EXPLOITATION & INSTALLATION		
EXTRACTION DU JOURNAL ARCHIVE VA PAC (DEXP)		10

10. EXTRACTION DU JOURNAL ARCHIVE VA PAC (DEXP)

10.1. PRESENTATION GENERALE

EXTRACTION DU JOURNAL ARCHIVE VA PAC : INTRODUCTION

La procédure DEXP extrait du fichier des mouvements archivés de VA Pac, les mouvements passés correspondant aux améliorations et les met en forme, afin de mettre à jour, dans la base DSMS, les éléments corrigés correspondant à chaque amélioration.

CONDITION D'EXECUTION

Aucune.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

REMARQUE

La procédure DEXP travaille avec un journal VA Pac \geq 2.0.

La procédure DEXQ travaille avec un journal VA Pac antérieur à la release 2.0.

10.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

Une ligne '*' (obligatoire) :

```
-----  
!Pos.! Lon.! Valeur  ! Signification      !  
!-----+-----+-----!  
!  2 !   1 !  '*'      ! Code carte          !  
!  3 !   8 ! uuuuuuuu ! Code utilisateur DSMS !  
! 11 !   8 ! pppppppp ! Mot de passe       !  
-----
```

Une ligne de demande d'extraction :

```
-----  
!Pos.! Lon.! Valeur ! Signification      !  
!-----+-----+-----!  
!  2 !   1 !  'J'      ! Code carte          !  
!    !   !   !          ! (obligatoire)     !  
!    !   !   !          ! LES ZONES SUIVANTES SONT FACULTATIVES !  
!  3 !   1 !  ' '      ! Liste des mouvements sélectionnés    !  
!    !   !  'N'      ! Pas de liste          !  
!  4 !  24 !          ! Sélection dans la base VA Pac :      !  
!  4 !   4 ! nnnn      ! Numéro de session: début de sélection !  
!  8 !   4 ! pppp      ! Numéro de session: fin de sélection  !  
!    !   !          ! -->(La sélection sur numéro(s) de    !  
!    !   !          ! session interdit celle sur dates)    !  
! 12 !   8 ! SSAAMMJJ ! Date de début de la sélection        !  
!    !   ! 'JOUR'    ! Date de début = date du jour         !  
! 20 !   8 ! SSAAMMJJ ! Date de fin de la sélection          !  
!    !   ! 'JOUR'    ! Date de fin = date du jour           !  
!    !   !          ! (par défaut si date de début='jour') !  
! 28 !   1 !          ! Version des mouvements sélectionnés  !  
!    !   !  ' '      ! Toutes les sessions                  !  
!    !   !  'T'      ! Sessions historisées de test         !  
!    !   !  'Z'      ! Session courante                     !  
! 29 !   3 ! ppp       ! Code produit                        !  
! 32 !   4 ! xxxx     ! Code logique base VA Pac             !  
! 36 !   3 ! bbb      ! Code bibliothèque                   !  
! 39 !  16 !          ! Type des entités à sélectionner     !  
! 55 !   1 !  ' '      ! Extraction des mouvements passés avec !  
!    !   !          ! l'amélioration 999999                !  
!    !   !  'N'      ! Mouvements de 999999 non extraits    !  
! 56 !   1 !  ' '      ! Edition des mouvements en double sur !  
!    !   !          ! une même entité VA Pac              !  
!    !   !  'N'      ! Mouvements en double non édités     !  
! 57 !   6 ! nnnnnn   ! Numéro d'amélioration                !  
-----
```

EDITION OBTENUE

Un compte rendu de l'extraction fournissant la liste des mouvements formatés.

RESULTAT OBTENU

Un fichier des mouvements de mise à jour de la base DSMS à reprendre en entrée de la procédure DUPT.

10.3. DESCRIPTION DES ETAPES

DEXP : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

EXTRACTION ET FORMATAGE DE MOUVEMENTS : PDS600

- . Fichiers permanents en entrée :
 - Fichier des données
 - PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7Dl\$\$SUF : DSN=&INDEX..&ROOT.&FILE.Dl
 - Fichier des libellés d'erreur
 - PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
 - Journal archivé VA Pac
 - PAC7PJ : DSN=&PAC7PJ
- . Fichier mouvement en entrée :
 - Mouvements utilisateurs
 - PACDMB : DSN=&&EXPJMB
- . Fichiers de tri
- . Fichier en sortie :
 - Mouvements de mise à jour DUPT
 - PACDMV : DSN=&&PACDMV
- . Etat en sortie :
 - Compte rendu demande de sélection
 - PACDRU
- . Codes retour :
 - . 0 : Pas d'erreur et pas de liste demandée
 - .04 : Pas d'erreur et édition de la liste des mouvements
 - .08 : Erreur sur la carte utilisateur ou paramètre
 - .12 : Erreur d'entrée/sortie sur un fichier

EDITION DES MOUVEMENTS DE MISE A JOUR : PDS610

- . Fichiers permanents en entrée :
 - Fichier des données
 - PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7Dl\$\$SUF : DSN=&INDEX..&ROOT.&FILE.Dl
 - Fichier des libellés d'erreur
 - PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
- . Fichier en entrée :
 - Mouvements de mise à jour de DSMS
 - PACDMV : DSN=&&PACDMV
- . Etat en sortie :
 - Liste des mouvements de mise à jour
 - PACDRU
- . Codes retour :
 - . 0 : Pas d'erreur
 - .12 : Erreur d'entrée/sortie sur un fichier

10.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* - EXTRACTION JOURNAL VA PAC - MAJ DSMS -
//*****
//$RADP.DEXP PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
// SPAMV='TRK,(2,1)', SPACE MOUVEMENTS EXTRAITS
// PAC7PJ='NULLFILE', DSN FICHER VA PAC MVTS ARCHIVES
// CYL=3, TAILLE WORKS DE TRI
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// SORTLIB='$BLIB', BIBLIOTHEQUE DE TRI
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* ENTREE UTILISATEUR
//* COL 02 : *
//* COL 03 : CODE UTILISATEUR DSMS
//* COL 11 : MOT DE PASSE
//* ENTREE LIGNE(S) COMMANDE(S) EXTRACTION
//* COL 02 : J
//* COL 03 : ' ' ----> LISTE DES MOUVEMENTS SELECTIONNES
//* : 'N' ----> PAS DE LISTE DES MOUVEMENTS SELECTIONNES
//* COL 04-07 : NUMERO DE SESSION DEBUT DE SELECTION
//* COL 08-11 : NUMERO DE SESSION FIN DE SELECTION
//* COL 12-19 : DATE DE DEBUT DE SELECTION (SSAAMMJJ)
//* COL 20-27 : DATE DE FIN DE SELECTION (SSAAMMJJ)
//* COL 28 : VERSION DE LA SESSION DES MOUVEMENTS SELECTIONNES
//* : ' ' ----> TOUTES LES SESSIONS
//* : 'T' ----> SESSIONS HISTORISEES DE TEST
//* : 'Z' ----> SESSION COURANTE
//* COL 29-31 : CODE PRODUIT
//* COL 32-35 : CODE BASE VA PAC INTERNE
//* COL 36-38 : CODE BIBLIOTHEQUE
//* COL 39-54 : TYPE DES ENTITES A SELECTIONNER
//* COL 55-55 : ' ' ----> EXTRAC MVTS PASSES AVEC L'AMELIO 999999
//* : 'N' ----> MOUVEMENTS DE 999999 NON EXTRAITS
//* COL 56-56 : ' ' ----> EDITION DES MVTS EN DOUBLE SUR UNE ENTITE
//* : VA PAC
//* : 'N' ----> MOUVEMENTS EN DOUBLE NON EDITES
//* COL 57-62 : NUMERO D'AMELIORATION
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&EXPJMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//
//VERIFY EXEC PGM=IDCAMS
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR

```

```

// *
//PDS600 EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,PDS600,PDS600$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDMB DD DSN=&&EXPJMB,DISP=(OLD,PASS)
//PACDMV DD DSN=&&PACDMV,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPAMV,RLSE),
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PAC7PJ DD DSN=&PAC7PJ,DISP=SHR
//PACDRU DD SYSOUT=&OUT
// *
//PDS610 EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,PDS610,PDS610$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
// COND=(4,NE,PDS600)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDMV DD DSN=&&PACDMV,DISP=(OLD,PASS)
//PACDRU DD SYSOUT=&OUT
// *

```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
EXTRACTION D'ENTITES

(DEXT)

PAGE 84

11

11. EXTRACTION D'ENTITES

(DEXT)

11.1. PRESENTATION GENERALE

EXTRACTION D'ENTITES (DEXT) : INTRODUCTION

Cette procédure permet l'extraction de toutes les entités DSMS sous forme de mouvements batch à utiliser en entrée de la procédure de mise à jour batch de la base DSMS (DUPT).

PRINCIPE

Cette procédure utilise, pour la sélection de l'extraction d'Améliorations, d'Evénements ou de Sites, des Requêtes (entités "Q") qui doivent donc avoir été préalablement définies dans la base DSMS. Ces trois types d'extraction doivent être demandés dans l'ordre précité.

Le code de la Requête utilisée doit être rappelé dans la demande d'extraction (voir les ENTREES UTILISATEUR).

La maquette (entité "R") affectée à la Requête utilisée pour l'extraction n'intervient pas dans l'extraction.

CONDITION D'EXECUTION

Aucune.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

Quelle que soit la cause de la fin anormale d'exécution, la procédure peut être relancée telle quelle, après suppression du problème.

11.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

Une ligne '*' (obligatoire) :

!Pos.!	Lon.!	Valeur	! Signification	!
! 2 !	! 1 !	! '*'	! Code carte	!
! 3 !	! 8 !	! uuuuuuuu	! Code utilisateur DSMS	!
! 11 !	! 8 !	! pppppppp	! Mot de passe	!
! 19 !	! 3 !	! ppp	! Code produit	!
! 22 !	! 2 !	! su	! Code filiale	!
! 24 !	! 1 !	! l	! Code langue	!

Il existe 4 types d'extraction possibles ; une ligne par demande est nécessaire :

!Pos.!	Lon.!	Valeur	! Signification	!
! 02 !	! 03 !	! 'PL'	! Verrouillage des bases	!
! 02 !	! 03 !	! Txx	! Codes de la table Txx	!
!	!	!	! (toutes tables sauf TRA)	!
! REQUETES / MAQUETTES :				!
! 02 !	! 04 !	! X QC	! Requête sur Améliorations	!
!	!	! X QE	! Requête sur Evénements	!
!	!	! X QS	! Requête sur Sites	!
! 02 !	! 04 !	! X RC	! Maquette sur Améliorations	!
!	!	! X RE	! Maquette sur Evénements	!
!	!	! X RS	! Maquette sur Sites	!
! 12 !	! 08 !	! uuuuuuuu	! Utilisateur propriétaire de la	!
!	!	!	! requête ou de la maquette	!
!	!	!	! (par défaut: utilisateur connecté)	!

!Pos.!	Lon.!	Valeur !	Signification	!
! 02 !	! 04 !	! LCQC !	! Requêtes sur Améliorations	!
! !	! !	! LCQE !	! Requêtes sur Evénements	!
! !	! !	! LCQS !	! Requêtes sur Sites	!
! 02 !	! 04 !	! LCRC !	! Maquettes sur Améliorations	!
! !	! !	! LCRE !	! Maquettes sur Evénements	!
! !	! !	! LCRS !	! Maquettes sur Sites	!
! 12 !	! 08 !	!uuuuuuuu!	! Utilisateur propriétaire des requêtes!	!
! !	! !	! !	! ou maquettes	!
! MOTS CLES :				!
! 02 !	! 04 !	! LAKC !	! Mots clés isolés des améliorations	!
! !	! !	! LGKC !	! Tous mots clés "	!
! 06 !	! 01 !	! 1 !	! Code langue des mots clés (langue de	!
! !	! !	! !	! l'utilisateur connecté par défaut)	!
! 02 !	! 04 !	! LAKE !	! Mots clés isolés natifs des Evénmts.	!
! !	! !	! LGKE !	! Tous mots clés "	!
! 02 !	! 04 !	! LAKT !	! Mots clés isolés techn. des Evénmts.	!
! !	! !	! LGKT !	! Tous mots clés "	!

```

+-----+
!Pos.! Lon.! Valeur ! Signification !
+-----+
!      !      !      ! .EXTRACTION PAR REQUETE UTILISATEUR :!
!  5 !  6 ! rrrrrr ! Code de la requête utilisateur (obli-!
!      !      !      ! gatoire) - Entité "Q" utilisée.    !
!  5 !  6 ! mmmmmm ! Code de la maquette (optionnel)    !
! 17 !  1 ! d      ! Délimiteur (optionnel)            !
!      !      !      ! Paramétrage :                      !
!      !      !      ! -----                          !
! 18 !  1 ! s      ! Symbole -                          !
! 19 !  1 ! x      ! Séparateur -                       !
! 20 ! 54 ! ..... ! Valeurs des paramètres -          !
!      !      !      !                                     !
!      !      !      ! Si des champs optionnels n'ont pas !
!      !      !      ! été renseignés, des valeurs par dé- !
!      !      !      ! faut sont utilisées. Elles provien- !
!      !      !      ! nent des lignes de définition de la !
!      !      !      ! requête de l'utilisateur trouvées  !
!      !      !      ! dans la base de données.          !
+-----+

```

EDITION OBTENUE

Un compte rendu d'extraction donnant le nombre de mouvements extraits.

RESULTAT OBTENU

Des mouvements de mise à jour de la base DSMS, à utiliser en entrée de la procédure DUPT.

Cette procédure renvoie un code retour général :

```

+-----+
!  0 ! OK !
!  8 ! Erreur sur ligne '*' !
!      ! ou sur ligne de commande !
! 12 ! ERREUR d'entrée-sortie ou !
!      ! base DSMS incohérente !
! 16 ! ERREUR de tri !
+-----+

```


11.3. DESCRIPTION DES ETAPES

DEXT : DESCRIPTION DES ETAPES

Cette procédure fait appel à un programme unique (PDSEX) qui sert de moniteur d'enchaînement des différents programmes, considérés comme sous-programmes de ce moniteur. Elle comprend les étapes suivantes :

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

EXTRACTIONS : PDSEX

- . Fichiers permanents en entrée :
 - Fichier des données
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
 - Fichier des éléments VisualAge Pacbase
PAC7DC\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
 - PAC7D3\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
 - Fichier des libellés d'erreur
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
- . Fichier mouvements en entrée :
 - Demandes d'extraction
PACDMB : DSN=&&PACDMB
- . Fichiers de travail :
 - Requêtes
PACDKQ
 - Fichiers temporaires
PACDW0, W1, W2, W3
PACDW4, W5, WI
- . Etats en sortie :
 - Compte-rendu d'enchaînement
PACDIA
 - Compte-rendu demandes extraction
PACDRU
- . Fichiers de tri
SORTWK01, 02, 03
- . Fichier en sortie :
 - Mouvements batch extraits
PACDIM : DSN=&&PACDIM

11.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5 *
//* - EXTRACTION MOUVEMENTS BATCH POUR DUPT - *
//*****
//$RADP.DEXT PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// LNG=, LANGAGE MONITEUR
// INDEX=' $INDEX ', INDEX VSAM
// INDEXP=' $INDEXP ', INDEX FICHIERS NON VSAM
//*: SYSTCAT=' $CATV ', CATALOGUE VSAM SYSTEME
//*: VSAMCAT=' $CATU ', CATALOGUE VSAM UTILISATEUR
// OUT=' $OUT ', CLASSE DE SORTIE
// OUTL=' $OUTL ', CLASSE DE SORTIE DES ETATS
// SPAMB=' TRK,(100,10)', SPACE COMMANDES EXTRACTION
// SPAIM=' TRK,(100,10)', SPACE DES MOUVEMENTS EXTRAITS
// SPAWK=' CYL,(020,02)', SPACE FICHIERS DE TRAVAIL
// CYL=3, TAILLE WORKS DE TRI
// STEPLIB=' $MODB ', BIBLIOTHEQUE LM BATCH
// SORTLIB=' $BIBT ', BIBLIOTHEQUE DE TRI
// PSBLIB=' $PSBLIB ', BIBLIOTHEQUE DES PSB
// DBDLIB=' $DBDLIB ', BIBLIOTHEQUE DES DBD
// DBDLIO=' $DBDLIO ', BIBLIOTHEQUE DES DBD VA PAC
// RESLIB=' $RESLIB ', RESLIB IMS
// PROCLIB=' $PRCLIB ', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40, SPIE=0, TEST=0, EXCPVR=0, RST=0, PRLD=, SRCH=0,
// CKPTID=, MON=N, LOGA=0, FMTO=T, DBRC=$DBRC, IRLM=$IRLM
//*-----*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPSCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSEX EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSEX&LNG,PDSEX$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*:STEPSCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)

```

```
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,  
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)  
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,  
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)  
//IMSMON   DD DUMMY  
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR  
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR  
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR  
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR  
//PAC7DC$$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR  
//PAC7D3$$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR  
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR  
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,DELETE)  
//PACDIA   DD SYSOUT=&OUTL  
//PACDKQ   DD UNIT=&UWK,SPACE=(&SPAMB),DCB=BLKSIZE=6160  
//PACDIM   DD DSN=&&PACDIM,DISP=(,PASS),  
//          UNIT=&UWK,SPACE=(&SPAIM),  
//          DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)  
//PACDRU   DD SYSOUT=&OUTL  
//PACDW0   DD UNIT=&UWK,SPACE=(&SPAMB),DCB=BLKSIZE=6160  
//PACDW1   DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6160  
//PACDW2   DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6080  
//PACDW3   DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6375  
//PACDW4   DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6080  
//PACDW5   DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6375  
//PACDWI   DD UNIT=&UWK,SPACE=(&SPAWK),DCB=BLKSIZE=6187  
//*
```

VisualAge Pacbase - Manuel d'Exploitation	PAGE	92
DSMS - EXPLOITATION & INSTALLATION		
EXTRACTION DE TABLES POUR LISTES EXTERNES (DEXH)		12

12. EXTRACTION DE TABLES POUR LISTES EXTERNES (DEXH)

12.1. PRESENTATION GENERALE

EXTRACTION DE TABLES POUR LISTES EXTERNES : INTRODUCTION

Cette procédure permet d'extraire l'ensemble des informations contenues dans les tables pour constituer un fichier, exploité sur le poste de travail du développeur. Ce fichier permet de créer les fichiers "Listes de valeurs externes", utilisés par les postes de travail DSMS dits "habillés", c'est-à-dire utilisant le module PAW.

Pour de plus amples informations, l'utilisateur se reportera au manuel "PAW GUIDE DU DEVELOPPEUR", chapitre "Habillage des produits IBM".

CONDITION D'EXECUTION

Aucune.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

Quelle que soit la cause de la fin anormale d'exécution, la procédure peut être relancée telle quelle, après suppression du problème.

12.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREE UTILISATEUR

```
-----  
!Pos.! Lon.! Valeur  ! Signification  !  
!-----+-----+-----+-----!  
!  2 !   1 ! '*'      ! Code carte      !  
!  3 !   8 ! uuuuuuuu ! Code utilisateur DSMS !  
! 11 !   8 ! pppppppp ! Mot de passe    !  
! 19 !   3 ! ppp      ! Code produit    !  
! 22 !   2 ! su       ! Code filiale    !  
-----
```

EDITION OBTENUE

Un compte rendu d'extraction avec la liste des tables extraites.

RESULTAT OBTENU

Toutes les tables générales (non liées à un produit) ainsi que les tables des OPTIONS, PHASES et VERSIONS du produit précisé dans l'entrée utilisateur.

12.3. DESCRIPTION DES ETAPES

DEXH : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

EXTRACTION DES TABLES POUR LISTES LOCALES : PDSXTH

Ce programme extrait les valeurs contenues dans les tables:
TST, TSU, TGR, TPR, TRE, TTY, TUD, TAT, TLA, TPH and TOP
pour consultation sur DSMS habillé.

- . Fichiers permanents en entrée :
 - Fichier des données
 - PAC7DA\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7D1\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
 - Fichier des libellés d'erreur
 - PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
- . Fichier mouvements en entrée :
 - Contrôle de l'utilisateur
 - PACDMB : DSN=&&PACDMB
- . Fichier en sortie :
 - Tables extraites
 - PACDMV : DSN=&&PACDMV
- . Etat en sortie :
 - Compte-rendu d'extraction
 - PACDRH
- . Fichiers de tri
 - SORTWK01, 02, 03

12.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* - EXTRACTION TABLES POUR LISTES EXTERNES -
//*****
//$RADP.DEXH PROC FILE=$FILE, NUMERO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME VA PAC
// INDEX=' $INDEX', INDEX VSAM
// INDEXP=' $INDEXP', INDEX DES FICHIERS NON VSAM
//* SYSTCAT=' $CATV', CATALOGUE SYSTEME VSAM
//* VSAMCAT=' $CATU', CATALOGUE UTILISATEUR VSAM
// OUT=' $OUT', CLASSE DE SORTIE
// OUTL=' $OUTL', CLASSE DE SORTIE DES ETATS
// SPAMV='TRK,(100,10)', 'SPACE' COMMANDES EXTRACTION
// CYL=3, TAILLE DES SORTWORKS
// STEPLIB=' $MODB', BIBLIOTHEQUE L.M. BATCH
// SORTLIB=' $BIBT', BIBLIOTHEQUE DE TRI
// PSBLIB=' $PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB=' $DBDLIB', BIBLIOTHEQUE DES DBD
// DBDLIO=' $DBDLIO', BIBLIOTHEQUE DES DBD VA PAC
// RESLIB=' $RESLIB', RESLIB IMS
// PROCLIB=' $PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//* DD DSN=&VSAMCAT,DISP=SHR
//SYS PRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSXTH EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSXTH,PDSXTH$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//* DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR

```


EXTRACTION DE TABLES POUR LISTES EXTERNES (DEXH)
JCL DE LA PROCEDURE

PAGE

97

12
4

```
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDMB DD DSN=&&PACDMB,DISP=(OLD,DELETE)
//PACDRH DD SYSOUT=&OUTL
//PACDMV DD DSN=&&PACDMV,DISP=(,PASS),
// UNIT=&UWK,SPACE=( &SPAMV),
// DCB=(RECFM=FB,LRECL=100,BLKSIZE=6200)
//*
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
MISE A JOUR BATCH

(DUPT)

PAGE 98

13

13. MISE A JOUR BATCH

(DUPT)

13.1. PRESENTATION GENERALE

MISE A JOUR BATCH (DUPT)

La procédure DUPT a pour fonction de mettre à jour les entités DSMS avec les mouvements issus des procédures DEXT, DEXP et/ou DXBJ.
Les mouvements peuvent également être directement entrés dans un fichier par l'utilisateur, sous éditeur. Pour la description détaillée des mouvements batch, reportez-vous à l'annexe "Structure des Mouvements Batch" du Manuel de Référence DSMS.

CONDITION D'EXECUTION

L'accès au TP DSMS doit être fermé.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

Quelle que soit la cause de la fin anormale d'exécution, la procédure peut être relancée telle quelle, après suppression du problème.

ATTENTION

DUPT est une procédure de mise à jour GLOBALE. Il faut donc bien remplir toutes les zones qui contiennent des données, sinon elles seront automatiquement remises à blanc.

Dans le cas d'une mise à jour batch sur un écran de définition d'une Amélioration, d'un Evènement ou d'un Site il y a deux lignes de mise à jour batch à remplir.

Bien que DSMS attribue automatiquement un numéro aux Améliorations et Evénements créés, lors de la création il est nécessaire d'attribuer un numéro provisoire. Par exemple, dans le cas de la création d'une Amélioration: C000001, où 000001 est le numéro provisoire que DSMS remplacera automatiquement par un numéro unique.

Il est par ailleurs impératif de renseigner le code action à C car aucune création implicite n'est effectuée.

Il est possible de créer plusieurs Améliorations et Evénements simultanément. Dans ce cas, le numéro provisoire de chaque Amélioration ou Evènement créé doit être unique. Exemple, pour la création simultanée de trois Améliorations : C000001, C000002 et C000003.

REMARQUE:

Un maximum de 2 520 améliorations et 2 520 événements est autorisé par flot de mouvements (limite interne du programme).

13.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

- . Une ligne de paramètres (facultative).
- . Une ligne d'identification par Produit/Filiale concernés par les mises à jour (obligatoire).
- . Les mouvements de mise à jour extraits et formatés par DEXT, DEXP ou DXBJ.
- . L'utilisateur doit ajouter au moins une ligne d'identification en face des mouvements de mise à jour.

Ligne de paramètres (facultative)

```

+-----+-----+-----+-----+
!Pos.! Lon.! Valeur  ! Signification                                     !
+-----+-----+-----+-----+
!  2 !  1 !  $      ! CODE LIGNE                                       !
!  3 !  1 !  !       ! TYPE DE TRAITEMENT / ORDRE DE TRI              !
!   !   !   !       ! Ce champ définit le mode de mise à           !
!   !   !   !       ! jour, ou le type de traitement à             !
!   !   !   !       ! utiliser pour l'ENSEMBLE des codes          !
!   !   !   !       ! utilisateur pour cette exécution           !
!   !   !   !       ! de la procédure batch DSMS.                 !
!   !   !   ! A      ! MODE DE MISE A JOUR NORMALE                  !
!   !   !   !       ! - Les mouvements sont triés en or-         !
!   !   !   !       ! dre ascendant, avant que toute             !
!   !   !   !       ! mise à jour soit appliquée (i.e.           !
!   !   !   !       ! les définitions d'entités sont             !
!   !   !   !       ! traitées avant les enregistre-            !
!   !   !   !       ! ments de sous-écrans).                     !
!   !   !   !       ! - Le 'mode de mise à jour' est             !
!   !   !   !       ! spécifié pour chaque enregistre-          !
!   !   !   !       ! ment d'identification.                     !
!   !   !   ! D      ! MODE ANNULATION                              !
!   !   !   !       ! - Les mouvements sont triés en or-         !
!   !   !   !       ! dre descendant avant que toute             !
!   !   !   !       ! mise à jour soit appliquée.                !
!   !   !   !       ! - Tous les mouvements sont traités        !
!   !   !   !       ! en annulation, de code action 'D'         !
!   !   !   !       ! - Les enregistrements d'identifica-       !
!   !   !   !       ! tion doivent spécifier le mode            !
!   !   !   !       ! 'NORMAL' - tout autre mode sera          !
!   !   !   !       ! considéré comme une erreur.                !
+-----+-----+-----+-----+

```

Pos.	Lon.	Valeur	Signification
4	1		INDICAT. DE FORMAT DE COMPTE-RENDU
		1	COMPTE-RENDU UNIQUE
			- Une ligne 'FIN DE COMPTE-RENDU' est créée.
			- Le 'NUMERO D'ENTREE' du mouvement est simplement incrémenté d'une unité pour chaque mouvement.
		2	IDENTIFICATION/CODE UTIL. FORMAT 1
			- Une ligne 'FIN DE COMPTE-RENDU' est créée pour chaque code utilisateur/enreg.d'identification.
			- Le 'NUMERO D'ENTREE' du mouvement est renseigné à 1 pour chaque enregistrement d'identification.
		3	IDENTIFICATION/CODE UTIL. FORMAT 2
			- Une ligne 'FIN DE COMPTE-RENDU' est créée pour chaque code utilisateur/enreg.d'identification.
			- Le 'NUMERO D'ENTREE' du mouvement est renseigné à 0 pour chaque enregistrement d'identification.
			L'enregistrement d'identification apparaîtra comme transaction numéro 0.

Si la ligne est absente, '\$A1' est pris en compte.

Format de la ligne d'identification (obligatoire)

!Pos.!	Lon.!	Valeur	! Signification	!
! 1 !	! 1 !	!	! CODE ACTION/MODE DE MISE A JOUR	!
!	!	!	! Cette zone définit le mode de mise	!
!	!	!	! à jour à mettre en oeuvre pour un	!
!	!	!	! code utilisateur donné.	!
!	!	!	!	!
!	!	! SPACE	! MODE NORMAL DE MISE A JOUR	!
!	!	!	! - Fonctionne comme DSMS en TP.	!
!	!	!	! - Si un évènement ou une améliora-	!
!	!	!	! tion est créé, tous les mouve-	!
!	!	!	! ments des sous-écrans suivants	!
!	!	!	! subiront la nouvelle modification!	!
!	!	! V	! MODE DE CONTROLE DE VERSION.	!
!	!	!	! - Tous les mouvements batch seront	!
!	!	!	! effectués avec un code action de	!
!	!	!	! création 'C'.	!
!	!	!	! - Les champs de références externes!	!
!	!	!	! concernant les définitions des	!
!	!	!	! évènements et des améliorations	!
!	!	!	! seront mis en place.	!
!	!	!	! - Le champ d'amélioration associé	!
!	!	!	! concernant la définition de	!
!	!	!	! l'évènement prendra le numéro de	!
!	!	!	! la 'nouvelle' amélioration -	!
!	!	!	! le numéro assigné lors de la	!
!	!	!	! création de l'amélioration.	!
!	!	! R	! MODE DE REORGANISATION	!
!	!	!	! - identique à 'V', excepté pour un	!
!	!	!	! point : le contenu des champs de	!
!	!	!	! référence externe ne sera pas	!
!	!	!	! modifié.	!

Pos.	Lon.	Valeur	Signification
2	1	*	CODE ENREGIST. D'IDENTIFICATION
3	8	...	UTILISATEUR DSMS
11	8	...	MOT DE PASSE DE L'UTILISATEUR
19	3	ppp	CODE DU PRODUIT concerné par les mises à jour.
22	2	ss	CODE DE LA FILIALE concernée par les mises à jour batch.
24	1	SPACE	Inutilisé
25	9		VALEURS DE REFERENCE EXTERNE
			La valeur des trois champs suivants permet de créer les références externes d'événements et d'améliorations, si le mode de mise à jour est 'V'.
25	4	dddA	- Code externe de la base DSMS
29	3	ppp	- Code produit externe DSMS.
32	2	ss	- Code filiale externe DSMS.
34	1		LIGNE BLANCHE APRES L'INDICATEUR D'ERREUR
		SPACE	Une ligne blanche sera imprimée après chaque message d'erreur, dans le compte-rendu.
		N	Aucune ligne blanche ne sera imprimée à la suite des messages d'erreur dans le compte-rendu.
35	1		INDICATEUR DE RUPTURE DE PAGE
		SPACE	Une rupture de page ne surviendra que lors d'un dépassement du nombre maximal de lignes par page.
		T	Saut de page à chaque nouveau type de mouvement.
		E	Une page pour chaque type de mouvement de chaque entité.
36	1		INDICATEUR DE TRI DES MOUVEMENTS
		SPACE	Les mouvements seront triés par type avant d'être traités
		N	Les mouvements seront traités dans leur ordre d'arrivée

EDITION OBTENUE

Un compte-rendu de mise à jour, avec des commentaires à propos des irrégularités et incohérences trouvées lors de l'exécution.

RESULTAT OBTENU

- . Une base DSMS prête à être utilisée en TP ou en batch,
- . Un fichier journal des mouvements ayant modifié la base, si la journalisation n'a pas été inhibée lors de la dernière restauration.

NOTE : Cette procédure incrémente le numéro de session s'il s'agit du premier accès à la base de la journée.

13.3. DESCRIPTION DES ETAPES

DUPT : DESCRIPTION DES ETAPES

VERIFICATION DES FICHIERS VSAM : IDCAMS

VERIFICATION INTEGRITE DE LA BASE : PDSBAS

. Fichiers permanents en entrée :
- Fichier des données
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- Fichier des libellés d'erreur
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Etat en sortie :
- Compte rendu de validité
PACDRS

. Code retour :

Cet utilitaire envoie un code retour 4 et provoque un ABEND en cas d'invalidité des données.

MISE A JOUR DE LA BASE DSMS : PDSUP0

. Fichiers permanents en entrée-sortie :
- Fichier des données
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- Fichier des éléments VisualAge Pacbase
PAC7DC\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
PAC7D3\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
- Fichier des références croisées
PAC7DX\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DX

. Fichier permanent en entrée :
- Fichier des libellés d'erreur
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Fichier mouvement en entrée :
- Mouvements de mise jour issus de DEXP
PACDIM : DSN=&INPUT

. Fichier en sortie :
-Journal
PAC7DJ\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DJ

. Etat en sortie :
- Compte-rendu de mise à jour
PACDRP

. Fichiers de tri :
SORTWK01, SORTWK02, SORTWK03

. Codes retour :
. 0 : Pas d'erreur détectée sur les fichiers
.08 : Erreur sur la carte utilisateur ou paramètre
.12 : Erreur d'entrée sortie sur un fichier

13.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5 *
//* MISE A JOUR DE LA BASE DSMS *
//*****
//$RADP.DUPT PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// CYL='(4,1)', TAILLE WORKS DE TRI
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// SORTLIB='$BIBT', BIBLIOTHEQUE DE TRI
// INPUT='NULLFILE', DSN MOUVEMENTS M.A.J. (DEXP,DEXT)
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// DBDLIO='$DBDLIO', BIBLIOTHEQUE DES DBD VA PAC
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR
//*
//PDSBAS EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSBAS,PDSBAS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDRS DD SYSOUT=&OUT

```

```
// *  
//PDSUP0 EXEC PGM=DFSRR00,REGION=$REGSIZ,  
// PARM=(DLI,PDSUP0,PDSUP0$$SUG,&BUF,  
// &SPIE&TEST&EXCPVR&RST,&PRLD,  
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)  
//STEPLIB DD DSN=&RESLIB,DISP=SHR  
// DD DSN=&STEPLIB,DISP=SHR  
//DFSRESLB DD DSN=&RESLIB,DISP=SHR  
//IMS DD DSN=&PSBLIB,DISP=SHR  
// DD DSN=&DBDLIB,DISP=SHR  
// DD DSN=&DBDLIO,DISP=SHR  
//*:STEP01 DD DSN=&SYSTCAT,DISP=SHR  
//*: DD DSN=&VSAMCAT,DISP=SHR  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//DDSNAP DD SYSOUT=&OUT  
//PROCLIB DD DSN=&PROCLIB,DISP=SHR  
//IEFRDER DD DUMMY,  
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)  
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,  
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)  
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,  
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)  
//IMSMON DD DUMMY  
//DFSVSAMP DD DSN=&INDEX..&ROOT.&ROOT.SY(DFSVSAMM),DISP=SHR  
//SORTLIB DD DSN=&SORTLIB,DISP=SHR  
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR  
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR  
//PAC7DC$$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR  
//PAC7D3$$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR  
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR  
//PAC7DJ$$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,DISP=SHR  
//PAC7DX$$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR  
//PACDIM DD DSN=&INPUT,DISP=SHR  
//PACDRP DD SYSOUT=&OUT  
// *
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
INITIALISATION

(DINI)

PAGE 109

14

14. INITIALISATION

(DINI)

14.1. PRESENTATION GENERALE

INITIALISATION (DINI) : INTRODUCTION

La procédure DINI a pour fonction d'initialiser les fichiers pour l'ouverture d'une nouvelle base DSMS.

Elle fournit une sauvegarde initiale des fichiers qui doit être installée par la procédure de restauration (DRST).

CONDITION D'EXECUTION

Aucune.

Cependant, les paramètres de la nouvelle base DSMS doivent avoir été préalablement choisis, différents de ceux d'autres bases DSMS existant éventuellement sur le site.

Les allocations et chargements initiaux des composants de la base DSMS (voir le déroulement de l'installation) doivent avoir été effectués.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre 'ANOMALIES D'EXECUTION' du chapitre "LES PROCEDURES BATCH".

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

14.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

La structure de l'entrée est la suivante :

```
-----  
!Pos.! Lon.! Valeur ! Signification !  
!-----+-----+-----+-----!  
! 2 ! 1 ! 'I' ! Code carte !  
! 3 ! 1 ! 'l' ! Code langue initiale !  
! ! ! ! (Option par défaut 'E' : anglais) !  
! 4 ! 1 ! ! NE RIEN RENSEIGNER SAUF POUR DOS/VSE !  
! ! ! 'I' ! option par défaut tout matériel !  
! ! ! 'N' ! si CURRENT-DATE = JJ/MM/AA en DOS/VSE!  
+-----+-----+-----+-----+
```

EDITION OBTENUE

Cette procédure édite un compte rendu énumérant les options mémorisées et le nombre d'enregistrements initiaux des fichiers de la base DSMS.

RESULTAT OBTENU

La sauvegarde initiale ainsi constituée contient :

- l'utilisateur initial '*****' mot de passe '*****' (Voir le paragraphe suivant : PREMIERE CONNEXION),
- une ligne dans la table des langues correspondant à la langue indiquée sur l'entrée utilisateur.

* IMPORTANT *

PREMIERE CONNEXION

Après exécution de la procédure DINI, exécuter la procédure de restauration DRST.

Une fois cette restauration correctement effectuée, la nouvelle base DSMS est installée.

Vérifier que la transaction TP d'accès à la nouvelle base est opérationnelle.

La première connexion à la base DSMS ainsi installée peut alors être effectuée, de la manière suivante :

- Entrer dans la transaction DSMS.
- Entrer sur l'écran initial le code utilisateur '*****' et le mot de passe '*****' et transmettre.
- Parmi les choix du menu, seuls ceux avec '*' sont accessibles. Ils correspondent aux tables à mettre à jour pour entrer les informations initiales indispensables au fonctionnement de DSMS. Ce sont, dans l'ordre :
 - la table des langues (choix TLA) : les codes et labels des langues utilisées,
 - la table des produits (choix TPR) : les codes et labels des produits,
 - la table des filiales (choix TSU) : les codes et labels des filiales,
 - la table des utilisateurs de DSMS : codes utilisateurs et autorisation d'accès (TUD, TUG, TUP et TUS).

(Voir le Manuel de Référence DSMS pour le détail de la gestion de ces tables).

Une fois les informations initiales entrées, le code utilisateur '*****' n'est plus nécessaire. Il ne peut pas être annulé, mais le responsable de DSMS sur le site peut interdire l'utilisation de ce code '*****' en lui attribuant un mot de passe secret, inconnu des autres utilisateurs.

14.3. DESCRIPTION DES ETAPES

DINI : DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

PRISE EN COMPTE DES ENTREES : PTU001

SAUVEGARDE INITIALE DE LA BASE : PDSINI

. Fichiers mouvement en entrée :

- Mouvement d'initialisation

PACDMB : DSN=&&DINIMB

. Fichier permanent en entrée :

- Libellés d'erreur

PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Fichier en sortie :

- Image séquentielle des fichiers

PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

. Etat en sortie :

- Compte rendu de sauvegarde

PACDRU

14.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* - INITIALISATION DES DSMS -
//*****
//$RADP.DINI PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
// INDEXP='$INDEXP', INDEX NON VSAM
// INDEXQ='$INDEXQ', INDEX FICHIERS A GENERATIONS
// OUT='$OUT', CLASSE DE SORTIE
// VOLS='SER=$VOLO', VOLUME FICHER GENERATION
// UNITS='$UNITS', UNITE FICHER GENERATION
// SPABB='TRK,(10,2)', SPACE DE LA SAUVEGARDE
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// UWK=$UWK, UNITE DE TRAVAIL
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* ENTREE
//* COL 02 : I
//* COL 03 : CODE LANGUE INITIAL (F=FRANCAIS, E=ANGLAIS)
//* COL 04 : FORMAT DATE MACHINE (N --> POUR JJ/MM/AA)
//* : (I --> POUR MM/JJ/AA)
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&DINIMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//PDSINI EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSINI,PDSINI$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//SYSPRINT DD SYSOUT=&OUT
//PACDMB DD DSN=&&DINIMB,DISP=(OLD,PASS)
//PACDDB DD DSN=&INDEXQ.&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=(&SPABB,RLSE),
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB
//PACDRU DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP.&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX.&ROOT.&ROOT.DE,DISP=SHR
//*

```

VisualAge Pacbase - Manuel d'Exploitation	PAGE	115
DSMS - EXPLOITATION & INSTALLATION		
EXTRACTION DU JOURNAL POUR MISE A JOUR (DXBJ)		15

15. EXTRACTION DU JOURNAL POUR MISE A JOUR (DXBJ)

EXTRACTION DU JOURNAL POUR MISE A JOUR	(DXBJ)	PAGE	116
PRESENTATION GENERALE			15
			1

15.1. PRESENTATION GENERALE

EXTRACTION DU JOURNAL DSMS (DXBJ) : INTRODUCTION

La procédure DXBJ extrait du fichier journal de DSMS les mouvements passés correspondant à un intervalle de date ou d'heure, ou à un utilisateur donné, pour les transformer en mouvements de mise à jour.

CONDITION D'EXECUTION

Aucune.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

15.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

Une ligne '*' (obligatoire) :

!Pos.!	Lon.!	Valeur	!	Signification	!
! 2 !	! 1 !	! '*'	!	! Code carte	!
! 3 !	! 8 !	! uuuuuuuu	!	! Code utilisateur DSMS	!
! 11 !	! 8 !	! pppppppp	!	! Mot de passe	!
!-----+-----+-----+-----+-----!					
! Facultatif					
!-----+-----+-----+-----+-----!					
! 19 !	! 3 !	! ppp	!	! Code produit	!
! 22 !	! 2 !	! su	!	! Code filiale	!
! 24 !	! 1 !	! 'F' ou 'E'	!	! Code langue	!
! !	! !	! !	!	! UTILISATEURS/MOTS DE PASSE DANS LES	!
! !	! !	! !	!	! MOUVEMENTS EN SORTIE :	!

Une ligne de demande d'extraction :

!Pos.!	Lon.!	Valeur	!	Signification	!
! 2 !	! 1 !	! 'K'	!	! Code carte	!
! 3 !	! 1 !	! ' '	!	! Liste des mouvements sélectionnés	!
! !	! !	! 'N'	!	! Pas de liste	!
! 4 !	! 8 !	! SSAAMMJJ	!	! Date de début de la sélection	!
! 12 !	! 8 !	! SSAAMMJJ	!	! Date de fin de la sélection	!
! 20 !	! 6 !	! HHMMSS	!	! Heure de début de la sélection	!
! 26 !	! 6 !	! HHMMSS	!	! Heure de fin de la sélection	!
! 32 !	! 8 !	! uuuuuuuu	!	! Code utilisateur sélectionné	!
! 40 !	! 1 !	! ' '	!	! Utilisateurs lus dans le journal	!
! !	! !	! !	!	! aucun mot de passe.	!
! !	! !	! 'T'	!	! Utilisateurs lus dans le journal	!
! !	! !	! !	!	! avec leur mot de passe si autorisa-	!
! !	! !	! !	!	! tion suffisante.	!
! !	! !	! 'l'	!	! Utilisateur et mot de passe précisés	!
! !	! !	! !	!	! sur les colonnes suivantes	!
! 41 !	! 8 !	! uuuuuuuu	!	! Utilisateur a reprendre dans les	!
! !	! !	! !	!	! mouvements en sortie (colonne 40=1)	!
! 48 !	! 8 !	! mmmmmmmm	!	! Mot de passe a reprendre dans les	!
! !	! !	! !	!	! mouvements en sortie (colonne 40=1)	!

EDITION OBTENUE

Un compte rendu de l'extraction et, sur demande, la liste des mouvements formatés

RESULTAT OBTENU

Un fichier des mouvements de mise à jour de la base DSMS à reprendre en entrée de la procédure DUPT. Un 'N' est positionné en colonne 36 des cartes utilisateur pour que DUPT ne trie pas ces mouvements.

15.3. DESCRIPTION DES ETAPES

DXBJ : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

EXTRACTION ET FORMATAGE DES MOUVEMENTS : PDS700

- . Fichiers permanents en entrée :
 - Fichier des données
 - PAC7DA\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7D1\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
 - Fichier des libellés d'erreur
 - PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
 - Journal DSMS archivé
 - PACDBJ : DSN=&INDEXQ..&ROOT.&FILE.BJ
- . Fichier mouvement en entrée :
 - Mouvements utilisateurs
 - PACDMB : DSN=&&DXBJMB
- . Fichier en sortie :
 - Fichier mouvements MAJ pour DUPT
 - PACDIM : DSN=&&PACDIM
- . Etats en sortie :
 - Compte rendu de l'extraction
 - PACDRK
 - Edition des mouvements
 - PACDSK
- . Codes retour
 - .0 : Pas d'erreur
 - .8 : Erreur sur la carte utilisateur ou paramètre
 - Absence à tort de la définition de l'environnement
 - .12: Erreur d'accès fichier
 - Enregistrement technique absent à tort

15.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* - EXTRACTION JOURNAL DSMS POUR MAJ DSMS -
//*****
//$RADP.DXBJ PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
// INDEXQ='$INDEXQ', INDEX FICHIERS A GENERATIONS
// SPAIM=(TRK,(015,05)), SPACE DES MVTS DE MAJ EXTRAITS
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* FORMAT DES MOUVEMENTS EN ENTREE :
//* .. UNE LIGNE UTILISATEUR ET MOT DE PASSE DSMS
//* COL 02 : *
//* COL 03 : CODE UTILISATEUR DSMS
//* COL 11 : MOT DE PASSE
//* COL 19 : CODE PRODUIT (FACULTATIF)
//* COL 22 : FILIALE (FACULTATIF)
//* COL 24 : CODE LANGUE (FACULTATIF)
//* .. LIGNE(S) COMMANDE(S) EXTRACTION
//* COL 02 : K
//* COL 03 : ' ' LISTE DES MOUVEMENTS SELECTIONNES
//* : 'N' PAS DE LISTE DES MOUVEMENTS SELECTIONNES
//* COL 04-11 : DATE DE DEBUT DE SELECTION (SSAAMMJJ)
//* COL 12-19 : DATE DE FIN DE SELECTION (SSAAMMJJ)
//* COL 20-25 : HEURE DE DEBUT DE SELECTION (HHMMSS)
//* COL 26-31 : HEURE DE FIN DE SELECTION (HHMMSS)
//* COL 32-39 : CODE UTILISATEUR
//* COL 40 : ' ' UTILISATEURS LUS DANS LE JOURNAL
//* : AUCUN MOT DE PASSE
//* : 'T' UTILISATEURS LUS DANS LE JOURNAL AVEC LEUR MOT
//* : DE PASSE SI AUTORISATION SUFFISANTE
//* : '1' UTILISATEUR ET MOT DE PASSE PRECISES SUR LES
//* : COLONNES SUIVANTES
//* COL 41-47 : CODE UTILISATEUR A REPRENDRE DANS LES MOUVEMENTS EN
//* : SORTIE (COLONNE 40=1)
//* COL 48-55 : MOT DE PASSE A REPRENDRE DANS LES MOUVEMENTS EN
//* : SORTIE (COLONNE 40=1)
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&DXBJMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX.&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX.&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX.&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFDE),DISP=SHR

```

```

// *
//PDS700 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDS700,PDS700$$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,, &DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// *:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// *: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOXX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDMB DD DSN=&&DXBJMB,DISP=(OLD,PASS)
//PACDIM DD DSN=&&PACDIM,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAIM,
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PACDBJ DD DSN=&INDEXQ..&ROOT.&FILE.BJ(0),DISP=SHR
//PACDRK DD SYSOUT=&OUT
//PACDSK DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
  
```


VisualAge Pacbase - Manuel d'Exploitation	PAGE	121
DSMS - EXPLOITATION & INSTALLATION		
MODIFICATION DE CODES ET DE MOTS CLES (DREN)		16

16. MODIFICATION DE CODES ET DE MOTS CLES (DREN)

16.1. PRESENTATION GENERALE

MODIFICATION DE CODES ET DE MOTS CLES (DREN) : INTRODUCTION

La procédure DREN permet de substituer de nouveaux codes (table ou site) ou mots clés à ceux définis et employés jusqu'alors dans les tables, thésaurus et autres entités les utilisant.

CONDITION D'EXECUTION

Cette procédure travaille à partir d'une sauvegarde séquentielle et/ou d'un journal archivé, elle doit donc être précédée d'une sauvegarde et/ou d'un archivage.

ANOMALIES D'EXECUTION

Se reporter au sous-chapitre "Anomalies d'Exécution" du chapitre "Les Procédures Batch".

Quelle que soit la cause de la fin anormale, la procédure peut être relancée telle quelle, après suppression du problème.

16.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREES UTILISATEUR

Une ligne '*' (obligatoire) :

!Pos.!	Lon.!	Valeur	! Signification
! 2 !	! 1 !	! '*'	! Code carte
! 3 !	! 8 !	! uuuuuuuu	! Code utilisateur
! 11 !	! 8 !	! pppppppp	! Mot de passe
! Facultatif			
! 19 !	! 3 !	! ppp	! Modifications effectuées sur les
! !	! !	! !	! entités dépendantes du code produit!
! !	! !	! !	! 'ppp'
! !	! !	! '***'	! Modifications effectuées sur les
! !	! !	! !	! entités dépendantes de tous les
! !	! !	! !	! codes produits
! 22 !	! 2 !	! ss	! Modifications effectuées sur les
! !	! !	! !	! entités dépendantes du code filiale!
! !	! !	! !	! 'ss'
! !	! !	! '***'	! Modifications effectuées sur les
! !	! !	! !	! entités dépendantes de tous les
! !	! !	! !	! codes filiales
! 24 !	! 1 !	! 'F' ou 'E'	! Code langue
! Obligatoire (au moins 1 des 2 zones à '1') !			
! 25 !	! 1 !	! ' '	! Pas de modification sur la
! !	! !	! !	! sauvegarde
! !	! !	! '1'	! Modifications sur la sauvegarde
! !	! !	! !	! !
! 26 !	! 1 !	! ' '	! Pas de modification sur l'archivage!
! !	! !	! '1'	! Modifications sur l'archivage

Lignes de commandes (500 maxi)

```

+-----+-----+-----+-----+
!Pos.! Lon.! Valeur ! Signification
+-----+-----+-----+-----+
!  2 !   3 ! 'Txx' ! choix table (idem TP)
!    !    ! 'Kxx' ! choix mot clé (avec xx = 'T ' pour
!    !    !      ! mots clés techniques, xx = 'E ' pour
!    !    !      ! mots clés natifs et xx = 'Cl' pour
!    !    !      ! mots clés de langue l d'amélioration)!
!    !    ! 'S ' ! choix site
!  5 !  13 !      ! ancien code
! 18 !  13 !      ! nouveau code
+-----+-----+-----+-----+

```

REMARQUES :

- Les codes (ancien et nouveau) doivent être précédés par 'C', 'E' ou 'S' pour la table TST, par 'C' ou 'E' pour les tables TGR, TTY et par 'F' ou 'R' pour la table TAT.
- Il n'est pas possible d'inverser deux codes (par exemple : 'AA' devient 'BB' et 'BB' devient 'AA') .En revanche, on peut renommer un code avec un nouveau (alors inconnu), et réutiliser ce premier code pour en transformer d'autres (par exemple : 'AA' devient 'BB' pendant que 'CC' et 'DD' deviennent 'AA'; il faudra dans ce cas écrire la commande AA/BB avant CC/AA et DD/AA).
- Les nouveaux codes attribués aux produits, filiales ou sites ne doivent pas déjà exister (dans la même filiale s'il s'agit du site).
- Les deux parties du code site (9 et 3 caractères) ne peuvent pas être modifiées séparément.
- Pour la table TVE, il est possible de demander les modifications suivantes :

- . Lot technique seul
- . Lot technique et release
- . Lot technique, release & matériel
- . Lot technique, release, matériel et version
(avec ou sans code langue)
- . Release seule
- . Matériel seul
- . Numéro de version (avec ou sans code langue)

Les morceaux isolés doivent être cadrés comme si le reste était présent. Des contrôles de cohérence vis à vis des ascendants sont effectués. Il faut tenir compte des modifications demandées sur les lignes précédentes.

- Le libellé associé au nouveau code peut être soit celui de l'ancien code, soit celui du code "écrasant", si celui-ci existait déjà. Ce choix est fait au moment du tri du fichier et est donc imprévisible.
- Pour les tables dépendantes d'un produit (TOP, TPH et TVE), le code de celui-ci doit être précisé (de façon explicite) sur la ligne "*".

EDITION OBTENUE

Compte rendu des modifications sur la sauvegarde et/ou l'archivage.

Remarque sur les compteurs :

Ils totalisent le nombre de mises à jour effectuées et non le nombre d'enregistrements modifiés (il peut y avoir plusieurs modifications sur un même enregistrement).

RESULTAT OBTENU

Si la modification a été effectuée sur l'archivage (1 en colonne 26), on obtient une nouvelle version de la sauvegarde séquentielle du journal.

Si la modification a été effectuée sur la sauvegarde de la base (1 en colonne 25), on obtient une nouvelle version de la sauvegarde séquentielle de la base qui doit être réorganisée par la procédure DREO avant d'être restaurée.

CODE RETOUR

```
+-----+
!  0  ! OK                                     !
!  8  ! Erreur sur ligne '*' ou sur ligne de commande !
! 10  ! Absence ou valeur erronée pour top sauvegarde !
! 11  ! Absence ou valeur erronée pour top archivage !
! 12  ! Erreur d'entrée-sortie ou base DSMS incohérente!
!    ! Absence à tort des tops sauvegarde/archivage !
! 16  ! Erreur de tri                           !
+-----+
```

16.3. DESCRIPTION DES ETAPES

DREN : DESCRIPTION DES ETAPES

Cette procédure fait appel à un programme unique (PDSMS) qui sert de moniteur d'enchaînement des différents programmes, considérés comme sous-programmes de ce moniteur. Elle comprend les étapes suivantes :

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

MODIFICATIONS : PDSMS

- . Fichiers permanents en entrée :
 - Fichier des données
 - PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
 - Fichier des libellés d'erreur
 - PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE
 - Fichier des références croisées
 - PAC7DX\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DX
 - Sauvegarde DSMS
 - PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(0)
 - Archivage DSMS
 - PACDBJ : DSN=&INDEXQ..&ROOT.&FILE.BJ(0)
- . Fichier en entrée :
 - Requêtes utilisateur
 - PACDMB : DSN=&&PACDMB
- . Fichiers de travail :
 - Demandes de modifications
 - PACDW0
 - Sauvegarde partielle triée
 - PACDW1
 - Sauvegarde partielle non triée
 - PACDW2
- . Fichiers en sortie :
 - Sauvegarde modifiée
 - PACDB3
 - Archive modifiée
 - PACDJB
- . Etats en sortie :
 - Compte-rendu d'enchaînement
 - PACDIA
 - Liste des commandes sur la sauvegarde
 - PACDIK
 - Compte-rendu des modifications (sauv)
 - PACDJK
 - Compte-rendu de la fusion (sauvegarde)
 - PACDIS
 - Liste des commandes sur l'archivage
 - PACDKK
 - Compte-rendu des modifications (arch)
 - PACDLK
- . Fichiers de tri
 - SORTWK01, 02, 03

16.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5 *
//* - CHANGEMENT CODES TABLE, MOTS CLES ET SITES - *
//*****
//$RADP.DREN PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// LNG=, LANGAGE MONITEUR
// INDEX=' $INDEX ', INDEX VSAM
// INDEXP=' $INDEXP ', INDEX FICHIERS NON VSAM
// INDEXQ=' $INDEXQ ', INDEX FICHIERS A GENERATION
//*: SYSTCAT=' $CATV ', CATALOGUE VSAM SYSTEME
//*: VSAMCAT=' $CATU ', CATALOGUE VSAM UTILISATEUR
// OUT=' $OUT ', CLASSE DE SORTIE
// SPAWK=' (TRK,(60,15)) ', SPACE POUR FICHIERS DA ET DC
// SPABB=' (TRK,(20,5),RLSE) ', SPACE SAUVEGARDE
// SPABJ=' (TRK,(20,5),RLSE) ', SPACE SAUVEGARDE
// CYL=3, TAILLE WORKS DE TRI
// VOLS=' $SER=$VOLO ', VOLUME FICHER GENERATION
// UNITS=' $UNITO ', UNIT FICHER GENERATION
// STEPLIB=' $MODB ', BIBLIOTHEQUE LM BATCH
// SORTLIB=' $BIBT ', BIBLIOTHEQUE DE TRI
// PSBLIB=' $PSBLIB ', BIBLIOTHEQUE DES PSB
// DBDLIB=' $DBDLIB ', BIBLIOTHEQUE DES DBD
// RESLIB=' $RESLIB ', RESLIB IMS
// PROCLIB=' $PRCLIB ', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSMS EXEC PGM=DFSRRC0,REGION=$REGSIZ,
// PARM=(DLI,PDSMS&LNG,PDSMS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)

```

```
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DX$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDMB   DD DSN=&&PACDMB,DISP=SHR
//PACDIA   DD SYSOUT=&OUT
//PACDIK   DD SYSOUT=&OUT
//PACDJK   DD SYSOUT=&OUT
//PACDKK   DD SYSOUT=&OUT
//PACDLK   DD SYSOUT=&OUT
//PACDIS   DD SYSOUT=&OUT
//PACDDB   DD DSN=&INDEXQ..&ROOT.&FILE.BB(0),DISP=OLD
//PACDBJ   DD DSN=&INDEXQ..&ROOT.&FILE.BJ(0),DISP=SHR
//PACDW0   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6160
//PACDW1   DD DSN=&&W1,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPAWK,
//          DCB=(RECFM=VB,BLKSIZE=6022,LRECL=354)
//PACDW2   DD DSN=&&W2,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPAWK,
//          DCB=(RECFM=VB,BLKSIZE=6022,LRECL=354)
//PACDB3   DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB
//PACDJB   DD DSN=&INDEXQ..&ROOT.&FILE.BJ(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABJ,
//          DCB=&INDEXQ..DSCB.&ROOT.&FILE.BJ
//*
//DEL10 EXEC PGM=IEFBR14,COND=(10,NE,PDSMS)
//*****
//DDBB DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(OLD,DELETE)
//DEL11 EXEC PGM=IEFBR14,COND=(11,NE,PDSMS)
//*****
//DDBJ DD DSN=&INDEXQ..&ROOT.&FILE.BJ(+1),DISP=(OLD,DELETE)
//DEL12 EXEC PGM=IEFBR14,COND=(12,NE,PDSMS)
//*****
//DDBB DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(OLD,DELETE)
//DDBJ DD DSN=&INDEXQ..&ROOT.&FILE.BJ(+1),DISP=(OLD,DELETE)
```


VisualAge Pacbase - Manuel d'Exploitation	PAGE	129
DSMS - EXPLOITATION & INSTALLATION		
PRE-PROCESSEUR DAF PROGRAMMES GENERES (DPDF)		17

17. PRE-PROCESSEUR DAF PROGRAMMES GENERES (DPDF)

17.1. PRESENTATION GENERALE

PRE-PROCESSEUR DAF DES PROGRAMMES GENERES : INTRODUCTION

La procédure DPDF traite les programmes générés utilisateur contenant des requêtes SQL d'accès à la Base par opérateurs DAF.

CONDITION D'EXECUTION

Aucune.

MISE EN OEUVRE

Cette procédure peut être exécutée de plusieurs manières :

- soit à la suite d'une génération de programmes par GPRT dont la sortie générée est reprise en entrée de la procédure DPDF avant envoi en compilation ou rangement dans une bibliothèque de programmes-sources,
- soit par appel de la procédure dans les cartes de contrôle optionnelles avant/après programme généré ; il faut alors avoir préalablement entré le JCL adéquat dans les options choisies, lesquelles sont mises à jour par la transaction de mise à jour des paramètres utilisateur ou par la procédure batch PARM.

17.2. ENTREES-TRAITEMENTS-RESULTATS

ENTREE UTILISATEUR

C'est le code source COBOL des programmes contenant des opérateurs DAF qui doivent être résolus par le pré-processeur avant compilation.

Chaque programme contient, après la ligne IDENTIFICATION DIVISION, une ligne de commande pour le pré-processeur :

```

-----
!Pos.! Lon.! Valeur  ! Signification
!-----+-----+-----+-----!
! 1 ! 6 ! nnnnnn ! Numéro de ligne COBOL
! 7 ! 1 ! '*' ! Commentaire
! 8 ! 5 ! 'TP ' ! Programme TP ou
! ! ! 'BATCH' ! Programme batch
! 13 ! 6 ! 'LIB:' ! Libellé fixe
! 19 ! 3 ! bbb ! Code bibliothèque
! 22 ! 1 ! blanc ! non utilisé
! 23 ! 5 ! nnnns ! Numéro session - Etat de session
! 28 ! 1 ! blanc ! non utilisé
! 29 ! 2 ! -- ! Variante(s) de génération
! 31 ! 5 ! 'AR:' ! Libellé fixe
! 36 ! 1 ! l ! Code langue de la base
! 37 ! 5 ! 'SC:' ! Squelette programmes Langage batch
! ! ! 'SG:' ! Squelette programmes Dialogue
! ! ! 'SR:' ! Squelette programmes Gén. Cobol
! 42 ! 1 ! l ! Langue du squelette
! 43 ! 1 ! blanc ! pas utilisé
! 44 ! 6 ! 'SINGLE' ! Simples quotes ou
! ! ! 'DOUBLE' ! Doubles quotes
-----

```

Exemples :

```
000020*TP LIB: APP 2345 00 AR: F SG: F SINGLE
```

```
000020*BATCH LIB: APP 2300T 4 AR: F SC: F DOUBLE
```

Cette ligne est générée automatiquement par la procédure GPRT.

EDITION OBTENUE

Un compte-rendu des erreurs éventuellement rencontrées.

RESULTAT OBTENU

Le résultat obtenu est le source COBOL dans lequel ont été résolus les opérateurs DAF et générés les appels aux sous-programmes batch ou TP d'accès à la Base.

17.3. DESCRIPTION DES ETAPES

DPDF : DESCRIPTION DES ETAPES

Cette procédure fait appel à un programme unique qui sert de moniteur d'enchaînement des différents programmes, considérés comme sous-programmes de ce moniteur. Elle comprend l'étape suivante :

PRISE EN COMPTE DES ENTREES : PTU001

VERIFICATION DES FICHIERS VSAM : IDCAMS

PRE-PROCESSEUR DU GENERE : DAFD10

. Fichiers permanents en entrée :
- Fichier des données
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- Fichier des libellés d'erreur
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Fichier en entrée :
- Programmes générés
DAF80 : DSN=&&DPDFMB

. Fichiers en sortie :
- Programmes générés à compiler
COB80 : DSN=&&DAFGEN

. Etats en sortie :
- Compte-rendu d'exécution
DAFREP

Remarque : si le flot généré contient les cartes de contrôle de compilation, dans le cas de l'utilisation de DPDF enchaînée derrière un GPRT, ce DSN (&&DAFGEN) peut être remplacé par l'envoi dans l'Internal Reader de la machine MVS de : SYSOUT=(&OUT,INTRDR).

17.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5 *
//* - ACCESS FACILITY PRE-PROCESSING - *
//*****
//$RADP.DPDF PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
// SPAMB='(TRK,(150,15))', SPACE PROGRAMMES GENERES
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// PSBLIB='$PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB='$DBDLIB', BIBLIOTHEQUE DES DBD
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&DPDFMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//DAFD10 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,DAFD10,PDSDAF$$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,, &DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//DAF80 DD DSN=&DPDFMB,DISP=(OLD,DELETE)
//COB80 DD DSN=&DAFGEN,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAMB,

```

PRE-PROCESSEUR DAF PROGRAMMES GENERES
JCL DE LA PROCEDURE

(DPDF)

PAGE

134

17
4

```
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)  
//DAFREP  DD SYSOUT=&OUT  
//*
```

VisualAge Pacbase - Manuel d'Exploitation	PAGE	135
DSMS - EXPLOITATION & INSTALLATION		
MISE A JOUR BATCH A PARTIR DE TABLES DAF (DUPD)		18

18. MISE A JOUR BATCH A PARTIR DE TABLES DAF (DUPD)

18.1. PRESENTATION GENERALE

DUPD : PRESENTATION GENERALE

La procédure DUPD effectue la mise à jour batch de la base DSMS à partir d'un fichier séquentiel à l'image des tables DAF.

Le principe de fonctionnement de cette procédure est tout à fait similaire à celui de la procédure DUPT, excepté le format des mouvements en entrée.

CONDITION D'EXECUTION

Se reporter au chapitre DUPT.

ANOMALIES D'EXECUTION

Se reporter au chapitre DUPT.

18.2. ENTREES-TRAITEMENTS-RESULTATS

DUPD : ENTREES - TRAITEMENTS - RESULTATS

ENTREES UTILISATEUR

Le fichier séquentiel des mouvements en entrée provient d'un programme extracteur DAF. Ses enregistrements sont à l'image des tables DAF, décrites dans le Manuel des TABLES DAF.

```
+-----+-----+-----+-----+
! Pos.! Lon.! Signification                                     !
!-----+-----+-----+-----+
!  1 !  1 ! Code mouvement (C, M, X, A ou D, B)                !
!  2 ! 10 ! Code de la table DAF                                !
! 12 ! 299 ! Contenu de la table DAF, telle qu'elle est       !
!      !    ! décrite dans le Manuel des TABLES DAF         !
+-----+-----+-----+-----+
```

REGLES DE MISE A JOUR

Les mouvements de mise à jour ne sont pas triés.
Chaque ensemble de mouvements affectant un produit ou une filiale doit être précédé d'une ligne de code table ASSIGN :

```
+-----+-----+-----+-----+
!Pos.!Lon.! Valeur   ! Signification                                     !
!-----+-----+-----+-----+
!  2 ! 10 ! 'ASSIGN' ! Code de la table                               !
! 12 !  8 ! uuuuuuuu ! Code utilisateur                               !
! 20 !  8 ! pppppppp ! Mot de passe                                   !
! 28 !  3 ! ppp      ! Code Produit                               !
! 31 !  2 ! ff       ! Code filiale                               !
+-----+-----+-----+-----+
```

EDITIONS OBTENUES

Se reporter au chapitre DUPT.

RESULTAT OBTENU

Se reporter au chapitre DUPT.

18.3. DESCRIPTION DES ETAPES

DUPD : DESCRIPTION DES ETAPES

VERIFICATION DES FICHIERS VSAM : IDCAMS

VERIFICATION INTEGRITE DE LA BASE : PDSBAS

. Fichiers permanents en entrée :
- Fichier des données
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- Fichier des libellés d'erreur
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Etat en sortie :
- Compte rendu de validité
PACDRS

. Code retour :

Cet utilitaire envoie un code retour 4 et provoque un ABEND en cas d'invalidité des données.

MISE EN FORME DES MOUVEMENTS : PDS900

. Fichier mouvement en entrée :
- Mouvements de mise à jour
PACDGY : DSN=&DAFINPUT Longueur=382

. Fichier en sortie :
- Mouvements mis en forme
PACDIM : DSN=&&PACDIM Longueur=250

MISE A JOUR DE LA BASE DSMS : PDSUP0

. Fichiers permanents en entrée-sortie :
- Fichier des données
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- Fichier des éléments VisualAge Pacbase
PAC7DC\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
PAC7D3\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
- Fichier des références croisées
PAC7DX\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DX

. Fichier permanent en entrée :
- Fichier des libellés d'erreur
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Fichier mouvement en entrée :
- Mouvements de mise jour
PACDIM : DSN=&&PACDIM

. Fichier en sortie :
- Journal
PAC7DJ\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DJ

. Etat en sortie :
- Compte-rendu de mise à jour
PACDRP

- . Fichiers de tri :
SORTWK01, SORTWK02, SORTWK03
- . Codes retour :
 - . 0 : Pas d'erreur détectée sur les fichiers
 - .08 : Erreur sur la carte utilisateur ou paramètre
 - .12 : Erreur d'entrée sortie sur un fichier

18.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* MISE A JOUR BASE DSMS A PARTIR DES TABLES DAF
//*****
//$RADP.DUPD PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// CYL='(4,1)', TAILLE WORKS DE TRI
// STEPLIB='$MODB', BIBLI LOAD-MODULES BATCH
// SORTLIB='$BIBT', BIBLI DE TRI
// DAFINPUT='NULLFILE', DSN MOUVEMENTS DE MISE A JOUR
// SPAIM='(TRK,(100,10),RLSE)', SPACE DES MOUVEMENTS
// PSBLIB='$PSBLIB', BIBLI DES PSB
// DBDLIB='$DBDLIB', BIBLI DES DBD
// DBDLIO='$DBDLIO', BIBLI DES DBD VA PAC
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR
//*
//PDSBAS EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSBAS,PDSBAS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDR DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR

```

```
//PACDRS DD SYSOUT=&OUT
//*
//PDS900 EXEC PGM=PDS900,COND=(0,NE,PDSBAS),REGION=0K
//*****
//*:STEPDCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACDGY DD DSN=&DAFINPUT,DISP=SHR
//PACDIM DD DSN=&&PACDIM,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAIM,
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=3500)
//*
//PDSUP0 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSUP0,PDSUP0$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,, &DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*:STEPDCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAMP),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DJ$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,DISP=SHR
//PAC7DX$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PACDIM DD DSN=&&PACDIM,DISP=(OLD,DELETE)
//PACDRP DD SYSOUT=&OUT
//*
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
INSTALLATION

PAGE 142

19

19. INSTALLATION

19.1. PREPARATION ENVIRONNEMENT

PREPARATION DE L'ENVIRONNEMENT

Il s'agit de préparer le site pour la mise en place du module DSMS, en s'appuyant sur les caractéristiques techniques du système, décrites dans les premiers chapitres de ce manuel :

- . faire le choix des divers préfixes suffixes et radicaux à affecter aux fichiers et codes DSMS,
- . faire le choix des catalogues VSAM et la réservation de l'espace disque nécessaire,
- . et surtout, préparer la génération d'IMS-DC en tenant compte des paramètres indiqués ci-dessous.

PARAMETRES POUR LA GENERATION D'IMS-DC :

Dans tous les noms précisés ci-après, les symboles yy zz et rr représentent deux suffixes et un radical choisis par l'utilisateur. Il faut cependant noter que l'implantation sera facilitée, en évitant la recompilation éventuelle des DBD et PSB, si les valeurs suivantes sont choisies :

```
zz = 22      pour le suffixe des PSB batch
yy = 22      pour le suffixe des DBD
rr = P2      pour le préfixe des PSB TP et
              transactions.
              (note: préfixe = radical du système)
```

1. Déclaration des DBD utilisés :

Nom des DBD :	PACDDAyy	Nom des DATASETS :	PAC7DAyy
	PACDDCyy (1)		PAC7DCyy
	PACDDEyy		PAC7DEyy
	PACDDJyy		PAC7DJyy
	PACDDXyy		PAC7DXyy
	PACDDZyy		PAC7DZyy
	PACDD1yy		PAC7D1yy
	PACDD3yy (1)		PAC7D3yy
	PACDDFyy		PAC7DFyy
	PACDF1yy		PAC7F1yy

```
DATABASE ACCESS=UP,DBD=(PACDDAyy, ...etc ... )
```

(1) : Ces DBSs sont livrés avec le système VA Pac.

2. Déclaration des transactions DSMS :
(conversationnelles)

a) APPLCTN PSB=rr00ZZ
 TRANSACT CODE=tttttttt,MSGTYPE=(SNGLSEG,RESPONSE),
 MODE=SNGL,SEGSIZE=02500,
 SPA=(150,DASD,FIXED),EDIT=ULC

avec tttttttt: code transaction pour la connexion
au module DSMS choisi pour le site.
(1 à 8 caractères).

b) APPLCTN PSB=rr0099
 TRANSACT CODE=rrT099,MSGTYPE=(SNGLSEG,RESPONSE),
 MODE=SNGL,SEGSIZE=02500,
 SPA=(150,DASD,FIXED),EDIT=ULC

NOTE : Il est fortement conseillé d'utiliser les transactions DSMS en
mode RESPONSE.

REMARQUE : Les sources des paramètres pour la génération d'IMS sont
livrées sur la bande dans le PDS des paramètres du système.

19.2. 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 module DSMS décrites dans ce manuel, afin de prévoir l'environnement nécessaire au bon déroulement de l'implantation (place sur disques, différents catalogues...).

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

PREPARATION

Récupération du JCL initial de la bande et exécution de celui-ci :

- . Sauvegarde de la bande d'implantation,
- . Copie d'un module de traitement du JCL complet,
- . Récupération du JCL complet d'implantation et d'exploitation de DSMS.

IMPLANTATION

(Voir sous-chapitre 'DEROULEMENT DE L'IMPLANTATION').

TESTS

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

19.3. CARTOUCHE (OU BANDE) D'INSTALLATION

CONTENU DE LA BANDE D'IMPLANTATION

La bande d'implantation (6250 BPI Labels Standards) contient les fichiers suivants :

! Rang	! Label	! Enreg.	! Bloc	! Contenu	!
! 01	! INST.JCL	! 80	! 11440	! JCL initial de	!
!	!	!	!	! préparation	!
! 02	! INST.MOD	!	! 6144	! Load-module MMLJCL,	!
!	!	!	!	! utilitaire de	!
!	!	!	!	! préparation du JCL	!
! 03	! PACD.JCL	! 80	! 11440	! JCL squelette	!
!	!	!	!	! d'implantation et	!
!	!	!	!	! d'exploitation	!
! 04	! PACD.MBR8	!	! 6144	! Load-modules batch	!
!	!	!	!	!	!
! 05	! PACD.MTR8	!	! 6144	! Load-modules TP	!
!	!	!	!	!	!
! 06	! PACD.DBDLIB	!	! 6144	! Fichier des DBDs	!
!	!	!	!	! objets	!
! 07	! PACD.PSBLIB	!	! 6144	! Fichier des PSBs	!
!	!	!	!	! objets	!
! 08	! PAC.BB	! 354	! 6376	! Sauvegarde de la	!
!	!	! (VB)	!	! base de tests	!
! 09	! PAC.DE	! 90	! 6300	! Fichier système des	!
!	!	!	!	! Libellés d'erreurs	!
! 10	! PACD.SOURCE	! 80	! 6080	! SOURCES ctl utilis.	!
!	!	!	!	! et dictionnaire DAF	!
! 11	! PACD.README	! 133	! 5320	! Fichier README	!
!	!	! (FBA)	!	! infos version	!

19.4. PREPARATION DE L'INSTALLATION

PREPARATION

Il est conseillé de ranger l'ensemble des JCLs de préparation, d'installation et d'exploitation de DSMS dans un fichier PDS réservé à cet effet.

La première phase de préparation consiste donc en l'allocation d'un tel fichier PDS, dont les caractéristiques sont les suivantes :

- Lrecl=80
- Taille : environ 30 pistes de disque 3390, 20 blocs directory.

(Cette allocation n'est pas fournie et doit être effectuée au préalable par la personne en charge de l'installation).

La deuxième phase consiste en la recopie du JCL initial de la cartouche ou bande d'implantation DSMS ('INST.JCL') par un utilitaire du site (IEBGENER par exemple), dans le PDS précédemment alloué.

JCL INITIAL

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

Les trois jobs du JCL initial à adapter à la configuration du site sont les suivants :

. RECOPIE DE LA CARTOUCHE (OU BANDE) D'INSTALLATION
livrée par IBM sur une cartouche de l'utilisateur : c'est cette dernière
cartouche, constituant la sauvegarde du module DSMS pour l'utilisateur,
qui doit être utilisée pour l'implantation.

. DECHARGEMENT DE L'UTILITAIRE MM1JCL, se trouvant dans le
fichier INST.MOD, par IEBCOPY dans une bibliothèque existant déjà
sur le site, ou allouée à cet effet.

Cette opération n'est à effectuer que pour la première implantation du
système.

. EXECUTION DE MM1JCL pour installer le JCL d'implantation et
d'exploitation de DSMS.

Compléter les lignes de JCL pour cette exécution :

```
//STEPLIB DD DSN= <-- librairie contenant MM1JCL  
//SYSUT1      <-- renseigner numéro de cartouche
```

```
//SYSUT2 DD DSN= <-- fichier récepteur du JCL complet  
d'installation-exploitation
```

Ce fichier peut être soit un membre du PDS prévu initialement pour
ranger l'ensemble des JCL, soit un fichier séquentiel choisi par
l'utilisateur.

Renseigner les paramètres (voir sous-chap. suivants).

CETTE EXECUTION DE MM1JCL DOIT ETRE CONSERVEE :
ELLE PEUT ETRE REUTILISEE POUR DES
REIMPLANTATIONS.

INSTALLATION
 PREPARATION DE L'INSTALLATION
 LE JCL INITIAL

19
 4
 1

19.4.1. LE JCL INITIAL

```
//***** DSMS IMS/V$ $REL $VV DU $DATE *****
//PACDSMS0 JOB (---), '3480', CLASS=D, MSGCLASS=A
//ALLOC EXEC PGM=IEHINITT
//TIBM DD DISP=SHR, UNIT=(3480, , DEFER), VOL=(, RETAIN, SER=$BDEIBM)
//TINST DD DISP=SHR, UNIT=(3480, , DEFER), VOL=(, RETAIN, SER=#####),
// DCB=DEN=3
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
TINST INITT SER=#####, OWNER='#####', DISP=REWIND
/*
//PACCOP PROC INDEX='$INPRO', NAME=XXX, LAB=N
//GENER EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=&INDEX. .&NAME, DISP=SHR,
// VOL=(, RETAIN, REF=* .ALLOC.TIBM), LABEL=&LAB
//SYSUT2 DD DSN=&INDEX. .&NAME, DISP=(, KEEP),
// VOL=(, RETAIN, REF=* .ALLOC.TINST), LABEL=&LAB,
// DCB=* .SYSUT1
// PEND
// *
//STEP01 EXEC PACCOP, LAB=01, NAME=JCL, INDEX=INST
//STEP02 EXEC PACCOP, LAB=02, NAME=MOD, INDEX=INST
//STEP03 EXEC PACCOP, LAB=03, NAME=JCL, INDEX=PACD
//STEP04 EXEC PACCOP, LAB=04, NAME=MBR8, INDEX=PACD
//STEP05 EXEC PACCOP, LAB=05, NAME=MTR8, INDEX=PACD
//STEP06 EXEC PACCOP, LAB=06, NAME=DBDLIB, INDEX=PACD
//STEP07 EXEC PACCOP, LAB=07, NAME=PSBLIB, INDEX=PACD
//STEP08 EXEC PACCOP, LAB=08, NAME=BB
//STEP09 EXEC PACCOP, LAB=09, NAME=DE
//STEP10 EXEC PACCOP, LAB=10, NAME=SOURCE, INDEX=PACD
//STEP11 EXEC PACCOP, LAB=11, NAME=README, INDEX=PACD
// *
//
```

INSTALLATION
PREPARATION DE L'INSTALLATION
LE JCL INITIAL

19
4
1

```
//PACDSMS1 JOB (---), 'UTI', CLASS=D, MSGCLASS=A
//*
//COPIE EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT3 DD UNIT=SYSDA, SPACE=(TRK,10)
//SYSUT4 DD UNIT=SYSDA, SPACE=(TRK,10)
//IM DD DISP=OLD, UNIT=3480, VOL=(, RETAIN, SER=#####),
// DSN=INST.MOD, LABEL=02
//OM DD DISP=SHR, DSN=###.###.###
//SYSIN DD *
C I=((IM,R)), O=OM
/*
/*
//
```

INSTALLATION
PREPARATION DE L'INSTALLATION
LE JCL INITIAL

19
4
1

```
//PACDSMS2 JOB (---),'JCL INSTALLATION',CLASS=D,MSGCLASS=A
//*
//ET010 EXEC PGM=IEFBR14
//DD1 DD DSN=fffffffffffffffffffff, <-- JOB PDS NAME/NOM DU PDS
// UNIT=%%%,VOL=SER=?????, <-- UNIT & VOLUME
// DISP=(,CATLG),
// SPACE=(TRK,(10,10,10)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//MM1JCL EXEC PGM=MM1JCL
//STEPLIB DD DISP=SHR,DSN=fff.fff.fff
//SYSOUT DD SYSOUT=A
//SYSUT1 DD DSN=PACD.JCL,DISP=OLD,
// UNIT=3480,VOL=(,RETAIN,SER=ffffff),LABEL=03
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT8 DD DUMMY,DCB=BLKSIZE=1370
//SYSUT9 DD DUMMY,DCB=BLKSIZE=1370
//*****
/* --- 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 DE CHAQUE MODULE
/* DE JCL.
//*****
/*
/* --- CREATION OF INSTALLATION JCL THROUGH PROGRAM 'MM1JCL' ----- *
//*****
/*
/* -MODIFY THE LIST OF THE SUPPLIED COMMANDS BY ENTERING THE
/* INSTALLATION VARIANTS (IF ANY), BY ASKING, IF NECESSARY, A
/* SELECTION OF PARTS OF INSTALLATION JCL (JCL MODULES), BY
/* GIVING THE APPROPRIATE VALUES TO THE INSTALLATION PARAMETERS*
/* AND BY SPECIFYING (OPTIONAL) THE LINES TO BE ADDED AT THE
/* BEGINNING OR AT THE END OF EACH JCL MODULE.
//*****
//SYSPRM DD DUMMY
//SYSUT2 DD DSN=&&PACDSMS2,DISP=(,PASS,DELETE),UNIT=ffffff,
// SPACE=(TRK,(10,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//SYSIN DD *
===SELL <> .LANGUAGE INSTALLATION (ENG OR FRA)
===SELV SEC .SECURITY SYSTEM INTERFACE
.INTERFACE SYSTEME DE SECURITE
===SELV DAF .DSMS ACCESS FACILITY
.MODULE DAF (DSMS ACCESS FACILITY)
===SELV ROLD .DSMS DATABASE RETRIEVAL
.REPRISE BASE DSMS
===PRM PRFJ=PAC .JOB NAMES PREFIX (3 CHARACTERS)
.PREFIXE DES NOMS DE JOBS (3 CARACTERES)
===PRM CCPT=<> .JOB ACCOUNTING CODES (JOB CARDS)
.CODE COMPTABLE DES JOBS (CARTES JOB)
===PRM CLASSJ=1 .JOB EXECUTION CLASS (JOB CARDS)
.CLASSE EXECUTION DES JOBS (CARTES JOB)
===PRM MSGCL=A .JCL OUTPUT CLASS (MSGCLASS)
.CLASSE DE SORTIE DU JCL (MSGCLASS)
===PRM UTAPE=3480 .CARTRIDGE UNIT
.UNITE DE CARTOUCHE
===PRM TAPEI=<> .INSTALLATION CARTRIDGE UNIT
.NOM DE LA CARTOUCHE D'INSTALLATION
===PRM OUT=A .JOB SYSOUT CLASS
.CLASSE DE SYSOUT DANS LES JOBS
===PRM INDEX='PAC25IMS.IMS' .DSMS SYSTEM VSAM FILE INDEXES
.INDEX DES FICHIERS VSAM
===PRM INDEXO='PAC25.SEQ' .VA PAC SYSTEM NO VSAM FILE INDEXES
.INDEX DES FICHIERS NON VSAM VA PAC
===PRM INDEXP='PAC25.SEQ' .DSMS SYSTEM NO VSAM FILE INDEXES
.INDEX DES FICHIERS NON VSAM
```

INSTALLATION
 PREPARATION DE L'INSTALLATION
 LE JCL INITIAL

19
 4
 1

```

===PRM INDEXQ='PAC25.SEQ' .DSMS GENERATION FILE INDEXES
                              .INDEX DES FICHIERS A GENERATION
===PRM ROOT='P2' .SYSTEM ROOT (2 CHARACTERS)
                              .RADICAL DU SYSTEME (2 CARACTERES)
===PRM ROOTO='P2' .VA PAC SYSTEM ROOT (2 CHARACTERS)
                              .RADICAL DU SYSTEME VA PAC (2 CARACTERES)
===PRM HEXA='D7F2' .SYSTEM ROOT (ROOT PARAM.) IN HEXAD.
                              .RADICAL DU SYSTEME (PRM ROOT) EN HEXA
===PRM FILE='00' .USER DATABASE NUMBER
                              .NUMERO DE BASE UTILISATEUR (2 CHIFFRES)
===PRM FILEO='00' .VA PAC USER DATABASE NUMBER
                              .NUMERO DE BASE UTILISATEUR VA PAC (2 CH)
===PRM SUF='22' .DBD SUFFIX (2 CHARACTERS)
                              .SUFFIXE DES DBD (2 CARACTERES)
===PRM SUG='22' .BATCH PSB SUFFIX (2 CHARACTERS)
                              .SUFFIXE DES PSB BATCH (2 CARACTERES)
===PRM RADP='D250' .PREFIX OF CATALOGUED PROCEDURE NAME
                              .PREFIXE NOMS DE PROCEDURES CATALOGUEES
===PRM REGSIZ='1536K' .BATCH PROCEDURE REGION SIZE
                              .TAILLE REGION POUR PROCEDURES BATCH
===PRM DBRC=N .DBRC USE IN BATCH PROCEDURE (N=NO)
                              .UTILISATION DBRC DANS PROC.BATCH (N=NON)
===PRM IRLM=N .IRLM USE IN BATCH PROCEDURE (N=NO)
                              .UTILISATION IRLM DANS PROC.BATCH (N=NON)
===PRM VOLV=<> .DSMS VSAM SYSTEM FILES VOLUME
                              .VOLUME DES FICHIERS DSMS SYSTEME VSAM
===PRM VOLU=<> .DSMS VSAM USER FILES VOLUME
                              .VOLUME DES FICHIERS DSMS UTILIS. VSAM
===PRM CATV=<> .VSAM CATALOG OF DSMS SYSTEM FILES
                              .CATALOGUE VSAM FICHIERS DSMS SYSTEME
===PRM CATU=<> .VSAM CATALOG OF DSMS USER FILES
                              .CATALOGUE VSAM FICHIERS DSMS UTILIS.
===PRM UWK=SYSDA .WORK UNIT
                              .UNITE DE TRAVAIL
===PRM UNITP=3390 .NO VSAM SYSTEM FILES UNIT
                              .UNITE DES FICHIERS SYSTEMES NON VSAM
===PRM UNITO=3390 .NO VSAM USER FILES UNIT
                              .UNITE DES FICHIERS UTILISATEURS NON VSAM
===PRM UNITV=3390 .VSAM SYSTEM FILES UNIT
                              .UNITE DES FICHIERS SYSTEMES VSAM
===PRM UNITU=3390 .VSAM USER FILES UNIT
                              .UNITE DES FICHIERS UTILISATEURS VSAM
===PRM VOLP=<> .NO VSAM SYSTEM FILES VOLUME
                              .VOLUME DES FICHIERS SYSTEMES NON VSAM
===PRM VOLO=<> .NO VSAM USER FILES VOLUME
                              .VOLUME FICHIERS UTILISATEURS NON VSAM
===PRM MODB='PAC25.PGMLIBBH' .DSMS LIBRARY OF BATCH MODULES
                              .BIBLI DSMS/MODULES BATCH
===PRM MODT='PAC25.PGMLIBTP' .DSMS LIBRARY OF ON-LINE MODULES
                              .BIBLI DSMS/MODULES TP
===PRM BIBP='SYS1.PROCLIB' .PROCEDURE LIBRARY
                              .BIBLIOTHEQUE DES PROCEDURES
===PRM BIBT='SYS1.SORTLIB' .SORT LIBRARY
                              .BIBLIOTHEQUE DE TRI
===PRM PSBLIB='IMSVS.PSBLIB' .PSB LIBRARY
                              .BIBLIOTHEQUE PSB
===PRM DBDLIB='IMSVS.DBDLIB' .DBD LIBRARY
                              .BIBLIOTHEQUE DBD
===PRM DBDLIO='IMSVS.DBDLIB' .VA PAC SYSTEM DBD LIBRARY
                              .BIBLIOTHEQUE DBD SYSTEME VA PAC
===PRM ACBLIB='IMSVS.ACBLIB' .ACB LIBRARY
                              .BIBLIOTHEQUE ACB
===PRM RESLIB='IMSVS.RESLIB' .IMS RESLIB
                              .RESLIB IMS
===PRM PRCLIB='IMSVS.PROCLIB' .IMS PROCLIB
                              .PROCLIB IMS
===PRM TRANS='DSMS' .TRANSACTION CODE FOR DSMS TRANSACTION
                              .CODE TRANSACTION CONNEXION A DSMS
===PRM CLS='2' .CLASS FOR TRANSACTION CODES
                              .CLASSE POUR LES CODES TRANSACTIONS
===BEGMOD
./ ADD NAME=$MODULE
/*

```


INSTALLATION
PREPARATION DE L'INSTALLATION
LE JCL INITIAL

19
4
1

```
//*  
//ET020 EXEC PGM=IEBUPDTE,PARM=NEW  
//SYSPRINT DD SYSOUT=X  
//SYSUT1 DD DSN=*.ET010.DD1,DISP=SHR  
//SYSUT2 DD DSN=*.ET010.DD1,DISP=SHR  
//SYSIN DD DSN=&&PACDSMS2,DISP=(OLD,DELETE)  
//
```

	PAGE	154
INSTALLATION		19
PREPARATION DE L'INSTALLATION		4
INSTALLATION DU JCL COMPLET		2

19.4.2. INSTALLATION DU JCL COMPLET

INSTALLATION DU JCL COMPLET

Le module MMIJCL lit le fichier de JCL squelette (label 03) et crée en sortie un JCL adapté à l'aide des commandes permettant:

- . de sélectionner la langue d'installation
- . de sélectionner des portions du JCL squelette, dénommées "modules de JCL",
- . de paramétrer le squelette de manière à obtenir un JCL avec un minimum de modifications à effectuer pour le rendre opérationnel,
- . de sélectionner des variantes d'installation pour générer le JCL propre à certains traitements spécifiques, en fonction du site ou des conditions d'installation,
- . de rajouter 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.

ENTREES UTILISATEUR :

Voir les paragraphes qui suivent :

- . Codification des commandes de MMIJCL
- . Tableau des variantes
- . Tableau des modules de JCL
- . Tableau des paramètres
- . Lignes en tête ou en fin de modules de JCL.

RESULTAT OBTENU : LE JCL COMPLET

Le fichier obtenu en SYSUT2 contient l'ensemble des JCLs d'installation et d'exploitation. Ce fichier doit être traité sous un éditeur de texte pour entamer l'installation proprement dite.

	PAGE	155
INSTALLATION		19
PREPARATION DE L'INSTALLATION		4
INSTALLATION DU JCL COMPLET		2

Deux manipulations sont à effectuer sur le JCL complet :

1. Modifications globales éventuelles :

Certaines adaptations peuvent être effectuées sur l'ensemble des JCLs.

Les CATALOGUES VSAM sont en commentaire dans le JCL obtenu à l'installation :

```

. dans les DELETE/DEFINE      : /*: CATALOG ($VCAT) */
                             /*: CATALOG ($SCAT) */
. dans les STEPCAT des JCL   : /*: STEPCAT DD
                             /*: DD
. dans les déclarations des
  paramètres des procédures : /*: VSAMCAT='$VCAT' ,
                             /*: SYSTCAT='$SCAT' ,

```

. Lorsque ces paramètres ne sont pas nécessaires sur le site, le JCL obtenu peut rester tel quel.

. Lorsque ces paramètres sont nécessaires sur le site, il faut alors rendre actives les lignes concernées en substituant ces commentaires, dans l'ordre :

- substitution de /*: par //
- substitution de /*: et de */ par du blanc.

Les FACTEURS DE BLOCAGE des fichiers de volume important peuvent aussi être modifiés. Voir le paragraphe "Remarque sur les Csize/Blksize des fichiers " (Va Pac seulement).

ATTENTION : SMS

. Dans les jobs d'installation contenant l'allocation de "Generationdatagroup", il est nécessaire de supprimer les cartes DD //GDGMOD des IDCAMS de définition.

. Si les paramètres UNIT et VOL sont interdits sur le site ceux-ci peuvent être supprimés par exclusion (EXCLUDE sous TSO/EDIT) dans l'ensemble du JCL.

De façon générale, il est conseillé d'effectuer toute autre modification d'ordre général sur les JCLs avant la manipulation suivante d'éclatement des JCLs.

	PAGE	156
INSTALLATION		19
PREPARATION DE L'INSTALLATION		4
INSTALLATION DU JCL COMPLET		2

2. Eclatement du fichier en différents membres :

En standard, le JCL complet contient devant chaque module une ligne ./ ADD NAME=<module-de-JCL>, où <module-de-JCL> est le code de la ligne ==MOD rencontrée (voir le Tableau des Modules de JCL ci-dessous).

Ceci permet l'éclatement du JCL complet obtenu en autant de membres que de modules de JCL dans un PDS. Pour cela, le fichier du JCL complet est à utiliser en SYSIN de l'utilitaire IEBUPDTE de mise-à-jour du PDS.

Remarque :

Du fait du choix de cette valeur par défaut, les ./ à l'intérieur des modules de JCL qui contiennent eux-mêmes l'utilitaire IEBUPDTE ont été remplacés par :/.

Une fois l'éclatement du JCL effectué, il faut resubstituer les :/ à l'intérieur des modules de JCL par des ./ lors de l'exécution des Jobs (jobs contenant des IEBUPDTE).

EDITION OBTENUE

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

Remarque : les paramètres du JCL squelette étant de la forme \$xxxx, lorsque MM1JCL rencontre à l'exécution un CARACTERE \$ ne correspondant pas à un paramètre défini, il envoie des messages d'erreur (par exemple, "PARAMETRE SYMBOLIQUE INCONNU", "ERREUR DE SYNTAXE DANS LE PARAMETRE SYMBOLIQUE", "POSITION OU LONGUEUR DU PARAMETRE SYMBOLIQUE ERRONEE"). Ces messages

Ces messages n'arrêtent pas le traitement et doivent être IGNORES : ils correspondent à des \$ dans le flot traité par MM1JCL qui ne sont effectivement pas des paramètres.

INSTALLATION
PREPARATION DE L'INSTALLATION
INSTALLATION DU JCL COMPLET

19
4
2

CODIFICATION DES COMMANDES DE MMLJCL

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

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

===SELM mmmml mmmm2Sélection de modules de JCL
mmml = nom de module de JCL
mmm2 = 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)

INSTALLATION	
PREPARATION DE L'INSTALLATION	
OPTIONS PAR DEFAUT INSTALLATION	

19
4
3

19.4.3. OPTIONS PAR DEFAUT INSTALLATION

VALEURS PAR DEFAUT :

.VARIANTES (===SELV) : toutes sont sélectionnées.

TRES IMPORTANT : SUPPRIMER LES LIGNES DES
VARIANTES NON ACQUISES PAR LE SITE OU LORSQU'IL Y A
INCOMPATIBILITE.

.PARAMETRES (===PRM) :

Les valeurs fournies sont des exemples. Il faut donc les renseigner en
fonction des choix du site.

.MODULES (===SELM) :

Aucune sélection de module n'est fournie. Tous les mo- dules
(correspondant aux variantes) sont sélectionnés.

.DEBUT DE MODULE DE JCL (===BEGMOD) :

```
./ ADD NAME=$MODULE
```

On obtient ainsi l'ajout d'une ligne devant chaque Module de JCL, de la forme

```
./ ADD NAME=<module-de-JCL>
```

INSTALLATION
PREPARATION DE L'INSTALLATION
VARIANTES D'INSTALLATION

PAGE

159

19
4
4

19.4.4. VARIANTES D'INSTALLATION

```
===SELV vvv .Commentaire
+-----+-----+
! vvv ! Signification !
+-----+-----+
! ! !
! DAF ! DSMS Access Facility !
! ! !
! ROLD ! Sélection de reprise de l'ancien DSMS !
! ! !
! SEC ! Interface systèmes de sécurité !
! ! Installation des sous-programmes d'accès au !
! ! système de sécurité (RACF ou TOPSECRET) !
! ! !
+-----+-----+
```

19.4.5. MODULES DE JCL

TABLEAU DES MODULES DE JCL : (==SELM mmmml mmmr2 ... mmmrn)

! mmm	! Contenu	! Nature !
! PACSLOAD	! Chargement des bibliothèques de programmes batch (MBR8) et TP (MTR8)	! JCL !
! PACSSY	! Chargement PDS 'SY' des paramètres	! JCL !
! PACSPROC	! Chargement des procédures Batch	! JCL !
! PACSDBD	! Chargement de la DBDLIB	! JCL !
! PACSPSB	! Chargement de la PSBLIB	! JCL !
! PACSPRE	! Allocations et chargements initiaux	! JCL !
! PACSDE	! Chargement de la base des libellés d'erreurs du système DSMS.	! JCL !
! PACSACB	! Compilation des ACB	! JCL !
! PACSRC	! Allocation/chargement PDS sources pour contrôles utilisateur	! JCL !
! PACSDS	! Restauration du jeu d'essai	! JCL !

TABLEAU DES MODULES DE JCL (suite)

! mmm	! Contenu	! Nature!
! JCLDARC	! Archivage du journal	! JCL !
! JCLDEXT	! Extraction (DEXT) et mise à jour éventuelle (DUPT)	! - !
! JCLDINS	! Liste des dates des programmes	! - !
! JCLDRST	! Restauration de la base	! - !
! JCLDSAV	! Sauvegarde de la base	! - !
! JCLDXBJ	! Extraction du journal sous forme de mouvements de mise à jour	! - !
! JCLDAF	! Exécution d'un programme batch d'accès à DSMS via DAF	! - !

19.4.6. PARAMETRAGE DU JCL

TABLEAU DES PARAMETRES :

```

===PRM PPPP=pppp                               .Commentaire

-----
! Code      ! Signification                               ! Défaut      !
! PPPP      !                                                    ! pppp        !
!-----!-----!-----!
!          ! Sur les cartes JOB                            !             !
!          !-----!
! PRFJ      ! Préfixe à 3 caractères des noms              ! PAC         !
!          ! des jobs                                     !             !
! CCPT      ! Code comptable des jobs                       ! <>          !
! CLASSJ    ! Classe d'exécution des jobs                   ! 1           !
! MSGCL     ! Classe de sortie du JCL                       ! A           !
!          !
!          ! Codification des DSN de fichiers            !             !
!          !-----!
!          ! Les fichiers permanents du module           !             !
!          ! DSMS (excepté les bibliothèques             !             !
!          ! de load-modules) ont des noms de           !             !
!          ! la forme:                                    !             !
!          !
!          ! INDEX.xxxxxss : Système VSAM                !             !
!          ! INDEXP.xxxxxss : Système non VSAM           !             !
!          ! INDEXX.xnnnss : Utilisateur VSAM            !             !
!          ! INDEXQ.xnnnss : Util. non VSAM              !             !
!          ! (GDG)                                        !             !
!          !
!          ! IND-- Index noms des fichiers :            !             !
! INDEX     ! Système et Utilisateur VSAM (2)! PAC25IMS.IMS !
! INDEXO    ! Système non VSAM du système                 ! PAC25.SEQ   !
!          ! VA Pac(SAM-PDS) (1)!
! INDEXP    ! Système non VSAM (SAM, PDS)                 ! PAC25.SEQ   !
! INDEXQ    ! Utilisateur non VSAM (GDG)                  ! PAC25.SEQ   !
!          !
!          ! xx=ROOT, nm=FILE, ss=code fich.           !             !
! ROOT      ! Radical du Système (2 caractères           !             !
!          ! différents de 'ZZ') (2)! P2
! ROOTO     ! Radical du Système VA Pac (carac-         !             !
!          ! tères différents de 'ZZ') (1)! P2
! FILE      ! Numéro base DSMS utilisateur               !             !
!          ! chiffres différents de 99) (2)! 00
! FILEO     ! Numéro de base VA Pac utilisateur         !             !
!          ! (2 chiffres différents de 99) (1)! 00
-----

```

TABLEAU DES PARAMETRES : (Suite)

```

-----
! Code      ! Signification                ! Défaut      !
! PPPP     !                               ! pppp        !
!-----!-----!-----!
!          !                               !             !
!          ! Sur les cartes DD            !             !
!          ! -----                    !             !
!          !                               !             !
! OUT       ! Classe d'impression des SYSOUTS ! A           !
! UTAPE     ! UNIT de la bande d'installation, ! 3480        !
!          ! copie de la bande livrée par IBM !             !
! TAPEI     ! Nom de la bande d'installation, ! <>          !
!          ! copie de la bande livrée par IBM !             !
! UWK       ! UNIT fich. de travail utilisés  ! SYSDA       !
! UNITP     ! UNIT fichiers Système non VSAM  ! 3390        !
! UNITO     ! UNIT fich. utilisateur non VSAM ! 3390        !
! UNITV     ! UNIT fichiers Système VSAM     ! 3390        !
! UNITU     ! UNIT fichiers utilisateur VSAM  ! 3390        !
! VOLP     ! Nom de volume fichiers Système ! <>          !
!          ! non VSAM                    !             !
! VOLV     ! Nom de volume fich. Système VSAM ! <>          !
! VOLO     ! Nom de volume fich. utilisateur ! <>          !
!          ! non VSAM                    !             !
! VOLU     ! Nom de volume fich. utilisateur ! <>          !
!          ! VSAM                        !             !
!          !                               !             !
!          ! Paramètres divers            !             !
!          ! -----                    !             !
!          !                               !             !
! RADP     ! Préfixe noms des procédures à  ! D250        !
!          ! installer (4 caractères maxi.) !             !
!          ! Ne modifier ce paramètre que si !             !
!          ! sa valeur par défaut normalement !             !
!          ! employée pour les noms de procé- !             !
!          ! dures DSMS ne convient pas     !             !
! HEXA     ! Radical du Système en hexadéci- ! D7F2        !
!          ! mal (voir paramètre ROOT).     !             !
! REGSIZ   ! Taille région procédures batch  ! 1536K       !
! DBRC     ! Utilisation de DBRC en batch    ! N           !
! IRLM     ! Utilisation de IRLM en batch    ! N           !
!          !                               !             !
-----

```

TABLEAU DES PARAMETRES : (Suite et fin)

```

-----
! Code      ! Signification                ! Défaut      !
! PPPP      !                               ! pppp        !
!-----!-----!-----!
!          !                               !             !
!          ! Paramètres divers (suite)    !             !
!          ! -----                      !             !
!          !                               !             !
! CATU      ! DSNNAME catalogue VSAM pour cata- ! <>          !
!          ! logage de la base DSMS de tests !             !
!          ! implantée à l'installation.    !             !
! CATV      ! DSNNAME catalogue VSAM sur lequel ! <>          !
!          ! les fichiers du module DSMS de  !             !
!          ! VA Pac sont à cataloguer.      !             !
! MODB      ! DSNNAME bibliothèque des Load-  ! PAC25.      !
!          ! modules batch DSMS.           ! PGMLIBBH   !
! MODT      ! DSNNAME bibliothèque des Load-  ! PAC25.      !
!          ! modules TP DSMS.              ! PGMLIBTP   !
! BIBP      ! DSNNAME bibli. des procédures  ! SYS1.PROCLIB !
!          ! dans laquelle sont à cataloguer !             !
!          ! les procédures DSMS           !             !
! BIBT      ! DSNNAME bibli. de tri utilisée  ! SYS1.SORTLIB !
! PSBLIB    ! DSNNAME bibliothèque des PSB    ! IMSVS.PSBLIB !
! ACBLIB    ! DSNNAME bibliothèque des ACB    ! IMSVS.ACBLIB !
! DBDLIB    ! DSNNAME bibliothèque des DBD    ! IMSVS.DBDLIB !
! DBDLIO    ! DSNNAME bibliothèque des DBD du  ! IMSVS.DBDLIB !
!          ! système VA Pac                 ! (1)!       !
! RESLIB    ! DSNNAME bibliothèque IMS RESLIB  ! IMSVS.RESLIB !
! PRCLIB    ! DSNNAME bibliothèque IMS PROCLIB ! IMSVS.PROCLIB !
! SUF       ! Suffixe des DBD DSMS (2)       ! 22         !
! SUG       ! Suffixe des PSB batch          ! 22         !
! CLS       ! Classe des codes transactions  ! 2          !
! TRANS     ! Code transaction de connexion au ! DSMS       !
!          ! module DSMS                    !           !
!          !                               !           !
-----

```

	PAGE	164
INSTALLATION		19
PREPARATION DE L'INSTALLATION		4
PARAMETRAGE DU JCL		6

(1) Les paramètres :

- 'ROOTO', 'FILEO', 'INDEXO' et 'DBDLIO' doivent OBLIGATOIREMENT avoir les MEMES VALEURS que celles attribuées aux paramètres respectifs : 'ROOT', 'FILE', 'INDEXP' et 'DBDLIB' du système VA Pac déjà implanté sur le site.

(2) Les paramètres :

- 'ROOT', 'FILE', 'SUF' et 'INDEX' doivent OBLIGATOIREMENT avoir les MEMES VALEURS que celles attribuées aux paramètres respectifs : 'ROOTD', 'FILED', 'SUFD' et 'INDEXD' du système VA Pac déjà implanté sur le site.

NOTE :

- . La valeur '<>' signale un paramètre devant obligatoirement être codé.
- . Les valeurs de paramètres contenant des caractères spéciaux doivent être délimitées par des apostrophes.

	PAGE	165
INSTALLATION		19
PREPARATION DE L'INSTALLATION		4
SEPARATEURS DES MODULES DE JCL		7

19.4.7. SEPARATEURS 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 MMIJCL 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 MMIJCL 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.

19.5. DEROULEMENT DE L'INSTALLATION

DEROULEMENT DE L'IMPLANTATION

Le JCL obtenu lors de la phase 2 se présente sous forme de 11 JOBS (chacun correspondant à un module de JCL) qui constituent l'implantation du système DSMS.

- 1 . Chargement des programmes batch et TP,
- 2 . Chargement du PDS des paramètres du système,
- 3 . Chargement des procédures batch d'exploitation,
- 4 . Chargement de la DBDLIB,
- 5 . Chargement de la PSBLIB,
- 6 . Préparation initiale des fichiers système,
- 7 . Chargement de la base des libellés d'erreurs du système DSMS,
- 8 . Compilation des ACB,
- 9 . Compléments d'installation,
- 10 . Chargement de la base de test,
- 11 . Edition de la liste des programmes installés avec leur date de compilation.

	PAGE	167
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
CHARGEMENT DES PROGRAMMES BATCH ET TP		1

19.5.1. CHARGEMENT DES PROGRAMMES BATCH ET TP

1. CHARGEMENT DES PROGRAMMES BATCH ET TP

Ce chargement est constitué d'un JOB '\$PRFJ.LOAD' qui comprend les étapes suivantes :

ET010 : IEHPROGM

SCRATCH UNCATLG des bibliothèques de load-modules
BATCH et TP.

ET020 : IEFBR14

Allocation bibliothèques des load-modules.

ET030 : IEBCOPY

Chargement des load-modules.

ET040 : IMASPZAP

ZAP du programme initiateur de la transaction

DSMS : \$ROOT.00ZZ.

A n'exécuter que si la valeur choisie pour le paramètre \$ROOT
n'est pas 'P2'.

ET050 : IEBCOPY

Chargement du sous-programme PACSECU8 d'accès aux
tables RACF dans une bibliothèque de programmes au_
torisée.

Il n'est pas nécessaire de définir de classe de res_
sources car les contrôles seront faits sur code uti_
lisateur et mot de passe uniquement.

(voir JCL ci-après.)

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

CHARGEMENT DES PROGRAMMES BATCH ET TP

1

```

//$PRFJ.LOAD JOB ($CCPT), 'LOAD-MODULES', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//* - CHARGEMENT PDS DES LOAD-MODULES BATCH ET TP - DSMS *
//*****
//*-----*
//*                A T T E N T I O N                *
//*                -----                            *
//* IL N'EST PAS NECESSAIRE D'EXECUTER L'ETAPE      *
//* 'ET040' SI LE PREFIX DES MODULES TP RESTE A    *
//* LA VALEUR 'P2' (CF PARAMETRE 'ROOT').          *
//* PAR CONTRE, IL EST OBLIGATOIRE DE L'EXECUTER  *
//* SI LE PREFIX EST MODIFIE. DANS CE CAS, IL     *
//* FAUT REMPLACER LE PARAMETRE 'HEXA' PAR LA    *
//* VALEUR CORRESPONDANT A CE NOUVEAU PREFIXE,   *
//* ET CODIFIEE EN HEXADECIMAL.                  *
//* EX: SI LE NOUVEAU PREFIXE EST: ROOT='AB',    *
//* ALORS LE PARAMETRE 'HEXA' DOIT ETRE         *
//* CODIFIE: HEXA='C1C2'.                        *
//*-----*
//* L'ETAPE 'ET050' NE CONCERNE QUE L'OPTION 'SEC' *
//* (CHARGEMENT -PACSECU8-)                       *
//*-----*
//*
//ET010 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD UNIT=$UNITP,VOL=SER=$VOLP,DISP=SHR
//SYSIN DD *
        UNCATLG DSN=$MODB
        SCRATCH DSN=$MODB,VOL=$UNITP=$VOLP
        UNCATLG DSN=$MODT
        SCRATCH DSN=$MODT,VOL=$UNITP=$VOLP
//*
//ET020 EXEC PGM=IEFBR14
//DDA DD DSN=$MODB,DISP=(,CATLG,DELETE),UNIT=$UNITP,
//      SPACE=(TRK,(55,10,15)),VOL=SER=$VOLP,
//      DCB=(RECFM=U,BLKSIZE=6144)
//DDB DD DSN=$MODT,DISP=(,CATLG,DELETE),UNIT=$UNITP,
//      SPACE=(TRK,(70,10,15)),VOL=SER=$VOLP,
//      DCB=(RECFM=U,BLKSIZE=6144)
//*
//ET030 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3 DD UNIT=$UWK,SPACE=(TRK,20)
//SYSUT4 DD UNIT=$UWK,SPACE=(TRK,20)
//INB DD DSN=PACD.MBR8,DISP=OLD,UNIT=$UTAPE,LABEL=(04,SL),
//      VOL=(,RETAIN,SER=$TAPEI)
//INT DD DSN=PACD.MTR8,DISP=OLD,UNIT=$UTAPE,LABEL=(05,SL),
//      VOL=(,RETAIN,SER=$TAPEI)
//OUTB DD DSN=$MODB,DISP=SHR
//OUTT DD DSN=$MODT,DISP=SHR
//SYSIN DD *
        COPY INDD=((INB,R)),OUTDD=OUTB
===SEQ FOR DAF
S M=DAFD10
S M=PDTPDF
===SEQ
S M=PDCHOI
S M=PDSA10
S M=PDSB
S M=PDSEX
S M=PDSMS
S M=PDSBE
S M=PDSEXE
S M=PDSMSE
S M=PDSUQ1
S M=PDSUQ2
S M=PDSUQ5
S M=PDSUQ6
S M=PDSUQ7
S M=PDSUQ8
S M=PDSRCT

```


INSTALLATION
DEROULEMENT DE L'INSTALLATION
CHARGEMENT DES PROGRAMMES BATCH ET TP

19
5
1

S M=PDSRFU
S M=PDSRMS
S M=PDSJMS
S M=PDSCAM
S M=PDSCEV
S M=PDSCMQ
S M=PDSCRQ
S M=PDSCSI
S M=PDSBAS
S M=PDSE90
S M=PDSEFAC
S M=PDSEINI
S M=PDSELVB
S M=PDSESRQ0
S M=PDSESRQ1
S M=PDSESRQ2
S M=PDSESRQ3
S M=PDSESR10
S M=PDSESR20
S M=PDSESR30
S M=PDSESR40
S M=PDSESUAA
S M=PDSESUB1
S M=PDSESUB2
S M=PDSESUB3
S M=PDSESUB4
S M=PDSEUE1
S M=PDSEUE2
S M=PDSEUE3
S M=PDSEUK1
S M=PDSEUSI
S M=PDSEUS1
S M=PDSEUS3
S M=PDSEUS4
S M=PDSEUS6
S M=PDSEUTT
S M=PDSEUTV
S M=PDSEUTW
S M=PDSEUTX
S M=PDSEUTY
S M=PDSEUTZ
S M=PDSEUT1
S M=PDSEUT2
S M=PDSEUT3
S M=PDSEUT4
S M=PDSEUT5
S M=PDSEUT6
S M=PDSEUT7
S M=PDSEUT8
S M=PDSEUT9
S M=PDSEUP0
S M=PDSEUP1
S M=PDSEV10
S M=PDSEV20
S M=PDSEXCT
S M=PDSEXDT
S M=PDSEXST
S M=PDSEXTH
S M=PDSEXTR
S M=PDSE300
S M=PDSE320
S M=PDSE380
S M=PDSE400
S M=PDSE450
S M=PDSE500
S M=PDSE600
S M=PDSE610
S M=PDSE700
S M=PDSETV090
S M=PDSE900
===SEQ FOR ROLD
S M=PDSESR8B
S M=PDSESR8C

INSTALLATION
DEROULEMENT DE L'INSTALLATION
CHARGEMENT DES PROGRAMMES BATCH ET TP

19
5
1

```
S M=PDSR8Q
S M=PDSR8R
S M=PDSR15
S M=PDSR5J
===SEQ
S M=REP2PJ
S M=PTU001
===SEQ FOR SEC
S M=PACSECB
===SEQ
COPY INDD=( ( INT,R ) ),OUTDD=OUTT
===SEQ FOR SEC
S M=PACSECB
===SEQ
S M=PACX10
S M=PACXAB
S M=MVSJOB
S M=( ( ZZR980,$ROOT.R980 ) )
S M=( ( ZZCHOI,$ROOT.CHOI ) )
S M=( ( ZZ00AA,$ROOT.00AA ) )
S M=( ( ZZ00AB,$ROOT.00AB ) )
S M=( ( ZZ00BA,$ROOT.00BA ) )
S M=( ( ZZ00B1,$ROOT.00B1 ) )
S M=( ( ZZ00B2,$ROOT.00B2 ) )
S M=( ( ZZ00B3,$ROOT.00B3 ) )
S M=( ( ZZ00B4,$ROOT.00B4 ) )
S M=( ( ZZ00B5,$ROOT.00B5 ) )
S M=( ( ZZ00EA,$ROOT.00EA ) )
S M=( ( ZZ00E1,$ROOT.00E1 ) )
S M=( ( ZZ00E2,$ROOT.00E2 ) )
S M=( ( ZZ00E3,$ROOT.00E3 ) )
S M=( ( ZZ00E4,$ROOT.00E4 ) )
S M=( ( ZZ00E5,$ROOT.00E5 ) )
S M=( ( ZZ00E6,$ROOT.00E6 ) )
S M=( ( ZZ00FA,$ROOT.00FA ) )
S M=( ( ZZ00FB,$ROOT.00FB ) )
S M=( ( ZZ00HE,$ROOT.00HE ) )
S M=( ( ZZ00JO,$ROOT.00JO ) )
S M=( ( ZZ00KA,$ROOT.00KA ) )
S M=( ( ZZ00K1,$ROOT.00K1 ) )
S M=( ( ZZ00K2,$ROOT.00K2 ) )
S M=( ( ZZ00K3,$ROOT.00K3 ) )
S M=( ( ZZ00LE,$ROOT.00LE ) )
S M=( ( ZZ00LS,$ROOT.00LS ) )
S M=( ( ZZ00MA,$ROOT.00MA ) )
S M=( ( ZZ00PA,$ROOT.00PA ) )
S M=( ( ZZ00P1,$ROOT.00P1 ) )
S M=( ( ZZ00QA,$ROOT.00QA ) )
S M=( ( ZZ00QB,$ROOT.00QB ) )
S M=( ( ZZ00QC,$ROOT.00QC ) )
S M=( ( ZZ00Q1,$ROOT.00Q1 ) )
S M=( ( ZZ00Q2,$ROOT.00Q2 ) )
S M=( ( ZZ00Q3,$ROOT.00Q3 ) )
S M=( ( ZZ00Q4,$ROOT.00Q4 ) )
S M=( ( ZZ00Q5,$ROOT.00Q5 ) )
S M=( ( ZZ00Q6,$ROOT.00Q6 ) )
S M=( ( ZZ00Q7,$ROOT.00Q7 ) )
S M=( ( ZZ00Q8,$ROOT.00Q8 ) )
S M=( ( ZZ00Q9,$ROOT.00Q9 ) )
S M=( ( ZZ00SA,$ROOT.00SA ) )
S M=( ( ZZ00SI,$ROOT.00SI ) )
S M=( ( ZZ00S1,$ROOT.00S1 ) )
S M=( ( ZZ00S3,$ROOT.00S3 ) )
S M=( ( ZZ00S4,$ROOT.00S4 ) )
S M=( ( ZZ00S5,$ROOT.00S5 ) )
S M=( ( ZZ00S6,$ROOT.00S6 ) )
S M=( ( ZZ00S7,$ROOT.00S7 ) )
S M=( ( ZZ00S8,$ROOT.00S8 ) )
S M=( ( ZZ00S9,$ROOT.00S9 ) )
S M=( ( ZZ00TA,$ROOT.00TA ) )
S M=( ( ZZ00TT,$ROOT.00TT ) )
S M=( ( ZZ00TU,$ROOT.00TU ) )
S M=( ( ZZ00TV,$ROOT.00TV ) )
```

INSTALLATION
 DEROULEMENT DE L'INSTALLATION
 CHARGEMENT DES PROGRAMMES BATCH ET TP

19
 5
 1

```

S M=((ZZ00TW,$ROOT.00TW))
S M=((ZZ00TX,$ROOT.00TX))
S M=((ZZ00TY,$ROOT.00TY))
S M=((ZZ00TZ,$ROOT.00TZ))
S M=((ZZ00T1,$ROOT.00T1))
S M=((ZZ00T2,$ROOT.00T2))
S M=((ZZ00T3,$ROOT.00T3))
S M=((ZZ00T4,$ROOT.00T4))
S M=((ZZ00T5,$ROOT.00T5))
S M=((ZZ00T6,$ROOT.00T6))
S M=((ZZ00T7,$ROOT.00T7))
S M=((ZZ00T8,$ROOT.00T8))
S M=((ZZ00T9,$ROOT.00T9))
S M=((ZZ00UD,$ROOT.00UD))
S M=((ZZ00ZZ,$ROOT.00ZZ))
S M=((ZZ0099,$ROOT.0099))
S M=((ZZCUAM,$ROOT.CUAM))
S M=((ZZCUEV,$ROOT.CUEV))
S M=((ZZCUMQ,$ROOT.CUMQ))
S M=((ZZCURQ,$ROOT.CURQ))
S M=((ZZCUSI,$ROOT.CUSI))
/*
//ET040 EXEC PGM=IMASPPAP
//SYSPPRINT DD SYSOUT=$OUT
//SYSLIB DD DSN=$MODT,DISP=SHR
NAME $ROOT.00ZZ P200ZZ
VER 0005 D7F2
REP 0005 $HEXA
VER 0275 D7F2
REP 0275 $HEXA
VER 02AB D7F2
REP 02AB $HEXA
/*
===SEQ FOR SEC
//ET050 EXEC PGM=IEBCOPY
//SYSPPRINT DD SYSOUT=$OUT
//SYSUT3 DD UNIT=$UWK,SPACE=(TRK,20)
//IND1 DD DSN=PACD.MBR8,DISP=OLD,UNIT=$UTAPE,
// VOL=(,RETAIN,SER=$TAPEI),LABEL=(04,SL)
//OUT1 DD DISP=OLD,DSN=----- <-- AUTHORIZED LIBRARY
COPY INDD=((IND1,R)),OUTDD=OUT1
S M=((PACSECRA,PACSECU8))
/*
//
===SEQ

```

	PAGE	172
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME		2

19.5.2. ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME

2. CHARGEMENT DU PDS DES PARAMETRES DU SYSTEME

Ce chargement est constitué d'un JOB '\$PRFJ.SY' qui comprend les étapes suivantes :

ET010 : IEHPROGM

SCRATCH UNCATLG du PDS des paramètres SY.

ET020 : IEFBR14

Allocation PDS des paramètres.

ET030 : IEBUPDTE

Chargement des paramètres, qui représentent:

- Les sources des DBD et PSB.
- Les définitions (DELETE/DEFINE) des fichiers VSAM.
- Les DELETE et LISTCAT de la base journal 'DJ'.
- Les vérifications (VERIFY) des fichiers VSAM.
- Le REPRO et l'enregistrement de chargement de la base des S.P.A. 'DZ'.
- Le REPRO et l'enregistrement de chargement du fichier de travail DF.
- Les définitions des buffers VSAM (DFSVSAMP).
- le membre PACSCTRL destiné à l'équipe système. Ce membre contient la description des macroinstructions 'APPLCTN' et 'TRANSACT' du système DSMS devant être définies dans la REGION DE CONTROLE IMS.

(voir JCL ci-après.)

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME

2

```

//SPRFJ.SY JOB ($CCPT),'LOAD PARAM.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//* CHARGEMENT DES PARAMETRES DE DSMS DANS LE PDS 'SY ' *
//*****
//*
//* *-----*
//* *           I M P O R T A N T           *
//* *           -----*
//* * AVANT D'EXECUTER CE JOB, REMPLACER GLOBALEMENT *
//* * A L'AIDE DE L'EDITEUR LES CARACTERES SUIVANTS: *
//* *           ':' PAR './' *
//* *-----*
//*
//* *-----*
//* *           N O T E           *
//* *           -----*
//* * CES PARAMETRES REPRESENTENT EN PARTICULIER LES *
//* * SYSINS D'ALLOCATION DES FICHIERS ET BASES CONS- *
//* * TITUTIFS DU MODULE DSMS. ADAPTER LES *
//* * TAILLES INDIQUEES DANS CES SYSINS A VOS BESOINS. *
//* *-----*
//*
//*
//ET010 EXEC PGM=IEHPRGM
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD UNIT=$UNITP,VOL=SER=$VOLP,DISP=SHR
//SYSIN DD *
        UNCATLG DSN=$INDEXP..$ROOT.$ROOT.SY
        SCRATCH DSN=$INDEXP..$ROOT.$ROOT.SY,VOL=$UNITP=$VOLP
//*
//ET020 EXEC PGM=IEFBR14
//DDA DD DSN=$INDEXP..$ROOT.$ROOT.SY,DISP=(,CATLG,DELETE),
//      UNIT=$UNITP,VOL=SER=$VOLP,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//      SPACE=(TRK,(03,02,04))
//*
//ET030 EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT1 DD DSN=$INDEXP..$ROOT.$ROOT.SY,DISP=SHR
//SYSUT2 DD DSN=$INDEXP..$ROOT.$ROOT.SY,DISP=SHR
//SYSIN DD *
:/ ADD NAME=PACDDA$SUF
    DBD NAME=PACDDA$SUF,ACCESS=(HIDAM,VSAM)
    DATASET DD1=PAC7DA$SUF,DEVICE=$UNITU,SIZE=4096
    SEGM NAME=PAC7DA,BYTES=(0350,0080)
    FIELD NAME=(CLEDA,SEQ,U),BYTES=40,START=03
    LCHILD NAME=(IPAC7D1,PACDD1$SUF),PTR=INDX
    DBDGEN
    END
:/ ADD NAME=PACDD1$SUF
    DBD NAME=PACDD1$SUF,ACCESS=(INDEX,VSAM)
    DATASET DD1=PAC7D1$SUF,DEVICE=$UNITU
    SEGM NAME=IPAC7D1,BYTES=40
    FIELD NAME=(CLED1,SEQ,U),BYTES=40,START=01
    LCHILD NAME=(PAC7DA,PACDDA$SUF),INDEX=CLEDA
    DBDGEN
    END
:/ ADD NAME=PACDDE$SUF
    DBD NAME=PACDDE$SUF,ACCESS=(HISAM,VSAM)
    DATASET DD1=PAC7DE$SUF,DEVICE=$UNITU,
    RECORD=98,SIZE=4096
    SEGM NAME=PAC7DE,BYTES=90
    FIELD NAME=(CLEDE,SEQ,U),BYTES=17,START=1
    DBDGEN
    END
:/ ADD NAME=PACDDJ$SUF
    DBD NAME=PACDDJ$SUF,ACCESS=(HDAM,OSAM),
    RMNAME=(CGIPACR1,20,16000000)
    DATASET DD1=PAC7DJ$SUF,
    DEVICE=$UNITU,SIZE=4096
    SEGM NAME=PAC7DJ,BYTES=187
    FIELD NAME=(CLEDJ,SEQ,U),BYTES=7,START=1

```

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME

2

```

        DBDGEN
        END
:/ ADD NAME=PACDDX$$SUF
        DBD      NAME=PACDDX$$SUF,ACCESS=(HISAM,VSAM)
        DATASET DD1=PAC7DX$$SUF,DEVICE=$UNITU,                $Y
                RECORD=88,SIZE=4096
        SEGM     NAME=PAC7DX,BYTES=80
        FIELD    NAME=(CLEDX,SEQ,U),BYTES=50,START=1
        DBDGEN
        END
:/ ADD NAME=PACDDZ$$SUF
        DBD      NAME=PACDDZ$$SUF,ACCESS=(HISAM,VSAM)
        DATASET DD1=PAC7DZ$$SUF,DEVICE=$UNITU,                $Y
                RECORD=9000,SIZE=10240
        SEGM     NAME=PAC7DZ,BYTES=8992
        FIELD    NAME=(CLEZ,SEQ,U),BYTES=12,START=1
        DBDGEN
        END
:/ ADD NAME=PACDDF$$SUF
        DBD      NAME=PACDDF$$SUF,ACCESS=(HIDAM,VSAM)
        DATASET DD1=PAC7DF$$SUF,DEVICE=$UNITU,SIZE=4096
        SEGM     NAME=PAC7DF,BYTES=(0550,0100)
        FIELD    NAME=(CLEDF,SEQ,U),BYTES=37,START=03
        LCHILD   NAME=(IPAC7F1,PACDF1$$SUF),PTR=INDX
        DBDGEN
        END
:/ ADD NAME=PACDF1$$SUF
        DBD      NAME=PACDF1$$SUF,ACCESS=(INDEX,VSAM)
        DATASET DD1=PAC7F1$$SUF,DEVICE=$UNITU
        SEGM     NAME=IPAC7F1,BYTES=37
        FIELD    NAME=(CLEF1,SEQ,U),BYTES=37,START=01
        LCHILD   NAME=(PAC7DF,PACDDF$$SUF),INDEX=CLEDF
        DBDGEN
        END
:/ ADD NAME=PDSB$$SUG
        PCB      TYPE=DB,DBDNAME=PACDDA$$SUF,PROCOPT=GOT,KEYLEN=40
        SENSEG   NAME=PAC7DA
        PCB      TYPE=DB,DBDNAME=PACDDC$$SUF,PROCOPT=GOT,KEYLEN=31
        SENSEG   NAME=PAC7DC
        PCB      TYPE=DB,DBDNAME=PACDDE$$SUF,PROCOPT=GOT,KEYLEN=17
        SENSEG   NAME=PAC7DE
        PSBGEN   PSBNAME=PDSB$$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PDSMS$$SUG
        PCB      TYPE=DB,DBDNAME=PACDDA$$SUF,PROCOPT=A,KEYLEN=40
        SENSEG   NAME=PAC7DA
        PCB      TYPE=DB,DBDNAME=PACDDC$$SUF,PROCOPT=A,KEYLEN=31
        SENSEG   NAME=PAC7DC
        PCB      TYPE=DB,DBDNAME=PACDDE$$SUF,PROCOPT=GOT,KEYLEN=17
        SENSEG   NAME=PAC7DE
        PCB      TYPE=DB,DBDNAME=PACDDJ$$SUF,PROCOPT=A,KEYLEN=07
        SENSEG   NAME=PAC7DJ
        PCB      TYPE=DB,DBDNAME=PACDDX$$SUF,PROCOPT=A,KEYLEN=50
        SENSEG   NAME=PAC7DX
        PSBGEN   PSBNAME=PDSMS$$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PDSBAS$$SUG
        PCB      TYPE=DB,DBDNAME=PACDDA$$SUF,PROCOPT=GOT,KEYLEN=40
        SENSEG   NAME=PAC7DA
        PCB      TYPE=DB,DBDNAME=PACDDE$$SUF,PROCOPT=GOT,KEYLEN=17
        SENSEG   NAME=PAC7DE
        PSBGEN   PSBNAME=PDSBAS$$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PDSEX$$SUG
        PCB      TYPE=DB,DBDNAME=PACDDA$$SUF,PROCOPT=GOT,KEYLEN=40
        SENSEG   NAME=PAC7DA
        PCB      TYPE=DB,DBDNAME=PACDDC$$SUF,PROCOPT=GOT,KEYLEN=31
        SENSEG   NAME=PAC7DC
        PCB      TYPE=DB,DBDNAME=PACDDE$$SUF,PROCOPT=GOT,KEYLEN=17
        SENSEG   NAME=PAC7DE
        PSBGEN   PSBNAME=PDSEX$$SUG,LANG=COBOL,CMPAT=YES
        END
:/ ADD NAME=PDSINI$$SUG

```

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME

2

```

PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PSBGEN   PSBNAME=PDSINI$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDSR10$SUG
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PSBGEN   PSBNAME=PDSR10$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDSR40$SUG
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PSBGEN   PSBNAME=PDSR40$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDSUP0$SUG
PCB      TYPE=DB, DBDNAME=PACDDA$SUF, PROCOPT=A, KEYLEN=40
SENSEGE  NAME=PAC7DA
PCB      TYPE=DB, DBDNAME=PACDDC$SUF, PROCOPT=A, KEYLEN=31
SENSEGE  NAME=PAC7DC
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PCB      TYPE=DB, DBDNAME=PACDDJ$SUF, PROCOPT=A, KEYLEN=07
SENSEGE  NAME=PAC7DJ
PCB      TYPE=DB, DBDNAME=PACDDX$SUF, PROCOPT=A, KEYLEN=50
SENSEGE  NAME=PAC7DX
PSBGEN   PSBNAME=PDSUP0$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDSV10$SUG
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=L, KEYLEN=17
SENSEGE  NAME=PAC7DE
PSBGEN   PSBNAME=PDSV10$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDSV20$SUG
PCB      TYPE=DB, DBDNAME=PACDDZ$SUF, PROCOPT=L, KEYLEN=12
SENSEGE  NAME=PAC7DZ
PSBGEN   PSBNAME=PDSV20$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDSXDT$SUG
PCB      TYPE=DB, DBDNAME=PACDDA$SUF, PROCOPT=GOT, KEYLEN=40
SENSEGE  NAME=PAC7DA
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PSBGEN   PSBNAME=PDSXDT$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDS300$SUG
PCB      TYPE=DB, DBDNAME=PACDDA$SUF, PROCOPT=AE, KEYLEN=40
SENSEGE  NAME=PAC7DA
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PCB      TYPE=DB, DBDNAME=PACDDJ$SUF, PROCOPT=GE, KEYLEN=07
SENSEGE  NAME=PAC7DJ
PSBGEN   PSBNAME=PDS300$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDS320$SUG
PCB      TYPE=DB, DBDNAME=PACDDA$SUF, PROCOPT=AE, KEYLEN=40
SENSEGE  NAME=PAC7DA
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PCB      TYPE=DB, DBDNAME=PACDDJ$SUF, PROCOPT=L, KEYLEN=07
SENSEGE  NAME=PAC7DJ
PSBGEN   PSBNAME=PDS320$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDS380$SUG
PCB      TYPE=DB, DBDNAME=PACDDE$SUF, PROCOPT=GOT, KEYLEN=17
SENSEGE  NAME=PAC7DE
PCB      TYPE=DB, DBDNAME=PACDDJ$SUF, PROCOPT=GE, KEYLEN=07
SENSEGE  NAME=PAC7DJ
PSBGEN   PSBNAME=PDS380$SUG, LANG=COBOL, CMPAT=YES
END
:/ ADD NAME=PDS400$SUG
PCB      TYPE=DB, DBDNAME=PACDDA$SUF, PROCOPT=L, KEYLEN=40
SENSEGE  NAME=PAC7DA
PCB      TYPE=DB, DBDNAME=PACDDC$SUF, PROCOPT=L, KEYLEN=31

```

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME

2

```

SENSEG NAME=PAC7DC
PCB TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG NAME=PAC7DE
PCB TYPE=DB,DBDNAME=PACDDJ$SUF,PROCOPT=L,KEYLEN=07
SENSEG NAME=PAC7DJ
PCB TYPE=DB,DBDNAME=PACDDX$SUF,PROCOPT=L,KEYLEN=50
SENSEG NAME=PAC7DX
PSBGEN PSBNAME=PDS400$SUG,LANG=COBOL,CMPAT=YES
END
:/ ADD NAME=PDS450$SUG
PCB TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=A,KEYLEN=40
SENSEG NAME=PAC7DA
PCB TYPE=DB,DBDNAME=PACDDC$SUF,PROCOPT=A,KEYLEN=31
SENSEG NAME=PAC7DC
PCB TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG NAME=PAC7DE
PCB TYPE=DB,DBDNAME=PACDDX$SUF,PROCOPT=A,KEYLEN=50
SENSEG NAME=PAC7DX
PSBGEN PSBNAME=PDS450$SUG,LANG=COBOL,CMPAT=YES
END
:/ ADD NAME=PDS500$SUG
PCB TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=AE,KEYLEN=40
SENSEG NAME=PAC7DA
PCB TYPE=DB,DBDNAME=PACDDC$SUF,PROCOPT=GE,KEYLEN=31
SENSEG NAME=PAC7DC
PCB TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG NAME=PAC7DE
PCB TYPE=DB,DBDNAME=PACDDX$SUF,PROCOPT=GE,KEYLEN=50
SENSEG NAME=PAC7DX
PSBGEN PSBNAME=PDS500$SUG,LANG=COBOL,CMPAT=YES
END
:/ ADD NAME=PDS600$SUG
PCB TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=GOT,KEYLEN=40
SENSEG NAME=PAC7DA
PCB TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG NAME=PAC7DE
PSBGEN PSBNAME=PDS600$SUG,LANG=COBOL,CMPAT=YES
END
:/ ADD NAME=PDS610$SUG
PCB TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=GOT,KEYLEN=40
SENSEG NAME=PAC7DA
PCB TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG NAME=PAC7DE
PSBGEN PSBNAME=PDS610$SUG,LANG=COBOL,CMPAT=YES
END
:/ ADD NAME=PDS700$SUG
PCB TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=GOT,KEYLEN=40
SENSEG NAME=PAC7DA
PCB TYPE=DB,DBDNAME=PACDDC$SUF,PROCOPT=GOT,KEYLEN=31
SENSEG NAME=PAC7DC
PCB TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG NAME=PAC7DE
PCB TYPE=DB,DBDNAME=PACDDJ$SUF,PROCOPT=GOT,KEYLEN=07
SENSEG NAME=PAC7DJ
PCB TYPE=DB,DBDNAME=PACDDX$SUF,PROCOPT=GOT,KEYLEN=50
SENSEG NAME=PAC7DX
PSBGEN PSBNAME=PDS700$SUG,LANG=COBOL,CMPAT=YES
END
:/ ADD NAME=PDSXTH$SUG
PCB TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=GOT,KEYLEN=40
SENSEG NAME=PAC7DA
PCB TYPE=DB,DBDNAME=PACDDC$SUF,PROCOPT=GOT,KEYLEN=31
SENSEG NAME=PAC7DC
PCB TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG NAME=PAC7DE
PCB TYPE=DB,DBDNAME=PACDDJ$SUF,PROCOPT=GOT,KEYLEN=07
SENSEG NAME=PAC7DJ
PCB TYPE=DB,DBDNAME=PACDDX$SUF,PROCOPT=GOT,KEYLEN=50
SENSEG NAME=PAC7DX
PSBGEN PSBNAME=PDSXTH$SUG,LANG=COBOL,CMPAT=YES
END
===SEQ FOR DAF
:/ ADD NAME=PDSDAF$SUG

```


INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME

2

```

PCB      TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=GOT,KEYLEN=40
SENSEG   NAME=PAC7DA
PCB      TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=GOT,KEYLEN=17
SENSEG   NAME=PAC7DE
PSBGEN   PSBNAME=PDSDAF$SUG,LANG=COBOL,CMPAT=YES
END
===SEQ
:/ ADD NAME=PTV090$SUG
PCB      TYPE=DB,DBDNAME=PACDDF$SUF,PROCOPT=L,KEYLEN=37
SENSEG   NAME=PAC7DF
PSBGEN   PSBNAME=PTV090$SUG,LANG=COBOL,CMPAT=YES
END
:/ ADD NAME=$ROOT.00ZZ
PCB      TYPE=TP,MODIFY=YES
PCB      TYPE=DB,DBDNAME=PACDDZ$SUF,PROCOPT=A,KEYLEN=12
SENSEG   NAME=PAC7DZ
PSBGEN   PSBNAME=$ROOT.00ZZ,LANG=COBOL
END
:/ ADD NAME=$ROOT.0099
PCB      TYPE=TP,MODIFY=YES
PCB      TYPE=DB,DBDNAME=PACDDA$SUF,PROCOPT=A,KEYLEN=40
SENSEG   NAME=PAC7DA
PCB      TYPE=DB,DBDNAME=PACDDC$SUF,PROCOPT=A,KEYLEN=31
SENSEG   NAME=PAC7DC
PCB      TYPE=DB,DBDNAME=PACDDE$SUF,PROCOPT=G,KEYLEN=17
SENSEG   NAME=PAC7DE
PCB      TYPE=DB,DBDNAME=PACDDJ$SUF,PROCOPT=A,KEYLEN=07
SENSEG   NAME=PAC7DJ
PCB      TYPE=DB,DBDNAME=PACDDX$SUF,PROCOPT=A,KEYLEN=50
SENSEG   NAME=PAC7DX
PCB      TYPE=DB,DBDNAME=PACDDZ$SUF,PROCOPT=A,KEYLEN=12
SENSEG   NAME=PAC7DZ
PSBGEN   PSBNAME=$ROOT.0099,LANG=COBOL
END
:/
ADD NAME=DF$ROOT.$FILE.D1
DELETE ($INDEX..$ROOT.$FILE.D1) CLUSTER
DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.D1)      -
                SHR (2,3)  RUS  KEYS (40 5)         -
                INDEXED                                     -
                VOL ($VOLUME)  TRK (10 5)           -
                RECSZ (046 046) )                   -
INDEX ( NAME ($INDEX..$ROOT.$FILE.D1.I)             -
        CISZ (1024) )                                -
DATA  ( NAME ($INDEX..$ROOT.$FILE.D1.D)             -
        CISZ (1024) ) /*: CATALOG ($CATU) /*/
:/
ADD NAME=DF$ROOT.$FILE.DA
DELETE ($INDEX..$ROOT.$FILE.DA) CLUSTER
DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.DA)      -
                SHR (2,3)  RUS                       -
                NONINDEXED                             -
                VOL ($VOLUME)  CYL (2 1)              -
                RECSZ (4089 4089) )                  -
DATA  ( NAME ($INDEX..$ROOT.$FILE.DA.D)             -
        FSPC (10,5)                                   -
        CISZ (4096) ) /*: CATALOG ($CATU) /*/
:/
ADD NAME=DF$ROOT.$ROOT.DE
DELETE ($INDEX..$ROOT.$ROOT.DE) CL
DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$ROOT.DE)      -
                SHR (2,3)  KEYS (17,6)               -
                VOL ($VOLUME) REC (70000)            -
                RECSZ (98,98) RUS )                  -
INDEX ( NAME ($INDEX..$ROOT.$ROOT.DE.I)             -
        CISZ (4096) )                                -
DATA  ( NAME ($INDEX..$ROOT.$ROOT.DE.D)             -
        FSPC (10,5)                                   -
        CISZ (4096) ) /*: CATALOG ($CATU) /*/
:/
ADD NAME=DF$ROOT.$FILE.DX
DELETE ($INDEX..$ROOT.$FILE.DX) CL
DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$FILE.DX)      -
                SHR (2,3)  KEYS (50 6)               -
                VOL ($VOLUME)  CYL (2 1)              -
                RECSZ (88 88) RUS )                  -
INDEX ( NAME ($INDEX..$ROOT.$FILE.DX.I)             -

```

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

ALLOCATION/CHARGEMENT PDS PARAMETRES SYSTEME

2

```

          CISZ (4096) ) -
DATA      ( NAME ($INDEX..$ROOT.$FILE.DX.D) -
          FSPC (5,5) -
          CISZ (4096) ) /*: CATALOG ($CATU) :*/
:/
  ADD NAME=DF$ROOT.$ROOT.DZ
DELETE ($INDEX..$ROOT.$ROOT.DZ) CLUSTER
DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$ROOT.DZ) -
                SHR (2,3) KEYS (12,6) -
                VOL ($VOLU) CYL (3 3) -
                RECSZ (9000,9000) RUS ) -
INDEX      ( NAME ($INDEX..$ROOT.$ROOT.DZ.I) -
          CISZ (4096) ) -
DATA      ( NAME ($INDEX..$ROOT.$ROOT.DZ.D) -
          FSPC (50,5) -
          CISZ (10240) ) /*: CATALOG ($CATU) :*/
:/
  ADD NAME=DF$ROOT.$ROOT.F1
DELETE ($INDEX..$ROOT.$ROOT.F1) CLUSTER
DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$ROOT.F1) -
                SHR (2,3) RUS KEYS (37 5) -
                INDEXED -
                VOL ($VOLU) TRK (10 5) -
                RECSZ (042 042) ) -
INDEX      ( NAME ($INDEX..$ROOT.$ROOT.F1.I) -
          CISZ (1024) ) -
DATA      ( NAME ($INDEX..$ROOT.$ROOT.F1.D) -
          CISZ (1024) ) /*: CATALOG ($CATU) :*/
:/
  ADD NAME=DF$ROOT.$ROOT.DF
DELETE ($INDEX..$ROOT.$ROOT.DF) CLUSTER
DEFINE CLUSTER ( NAME ($INDEX..$ROOT.$ROOT.DF) -
                SHR (2,3) RUS -
                NONINDEXED -
                VOL ($VOLU) CYL (5 1) -
                RECSZ (4089 4089) ) -
DATA      ( NAME ($INDEX..$ROOT.$ROOT.DF.D) -
          FSPC (10,5) -
          CISZ (4096) ) /*: CATALOG ($CATU) :*/
:/
  ADD NAME=DL$ROOT.$FILE.DJ
SCRATCH DSNAME=$INDEX..$ROOT.$FILE.DJ,VOL=$UNITU=$VOLU
UNCATLG DSNAME=$INDEX..$ROOT.$FILE.DJ
:/
  ADD NAME=VERIFDA
VERIFY FILE (DDDA)
:/
  ADD NAME=VERIFD1
VERIFY FILE (DDD1)
:/
  ADD NAME=VERIFDC
VERIFY FILE (DDDC)
:/
  ADD NAME=VERIFD3
VERIFY FILE (DDD3)
:/
  ADD NAME=VERIFDE
VERIFY FILE (DDDE)
:/
  ADD NAME=VERIFDX
VERIFY FILE (DDDX)
:/
  ADD NAME=VERIFDZ
VERIFY FILE (DDDZ)
:/
  ADD NAME=VERIFDF
VERIFY FILE (DDDF)
:/
  ADD NAME=VERIFF1
VERIFY FILE (DDF1)
:/
  ADD NAME=REPRODZ
REPRO INFILE (INDZ) OUTFILE (OUTDZ)
:/
  ADD NAME=REPRODF
REPRO INFILE (INDF) OUTFILE (OUTDF)
:/
  ADD NAME=MAXKEY
999999999999
:/
  ADD NAME=REPRO999
REPRO INFILE (MAXKEY) OUTFILE (SYSPAF)
:/
  ADD NAME=LI$ROOT.$FILE.DJ
LISTCAT ENTRIES ($INDEX..$ROOT.$FILE.DJ)
:/
  ADD NAME=DFSVSAM8
4096,8
:/
  ADD NAME=DFSVSAM9
12288,9
:/
  ADD NAME=DFSVSAMB
4096,8

```


	PAGE	180
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
CHARGEMENT DES PROCEDURES BATCH		3

19.5.3. CHARGEMENT DES PROCEDURES BATCH

3. CHARGEMENT DES PROCEDURES D'EXPLOITATION

Ce chargement est constitué d'un JOB '\$PRFJ.PROC' qui effectue, par IEBUPDTE, la mise en catalogue dans la PROCLIB de toutes les procédures batch d'exploitation.

Chaque procédure est un membre codifié '\$radp.NNNN', où radp est le radical choisi à la génération du JCL et NNNN le nom standard de la procédure DSMS. Les procédures sont décrites dans les autres chapitres du présent manuel.

(voir JCL ci-après.)

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

CHARGEMENT DES PROCEDURES BATCH

3

```

//$PRFJ.PROC JOB ($CCPT), 'LOADING PROCEDURES', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//ET010 EXEC PGM=IEBUPDTE, PARM=NEW
//*****
//*          CATALOGAGE DES PROCEDURES DSMS          *
//*****
//*
//* *-----*
//* *          I M P O R T A N T          *
//* *          -----          *
//* * AVANT D'EXECUTER CE JOB, REMPLACER GLOBALEMENT *
//* * A L'AIDE DE L'EDITEUR LES CARACTERES SUIVANTS: *
//* *          ':' PAR './'          *
//* *-----*
//*
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2 DD DSN=$BIBP, DISP=SHR
//SYSIN DD DATA, DLM='%%'
:/ ADD NAME=$RADP.DARC
//* DSMS : ARCHIVAGE DU JOURNAL *
:/ ADD NAME=$RADP.DEXH
//* DSMS : EXTRACTION DES TABLES POUR LISTES EXTERNE *
:/ ADD NAME=$RADP.DEXP
//* DSMS : EXTRACTION JOURNAL VA PAC - MAJ DSMS *
:/ ADD NAME=$RADP.DEXT
//* DSMS : EXTRACTION MOUVEMENTS BATCH POUR 'DUPT' *
:/ ADD NAME=$RADP.DINI
//* DSMS : INITIALISATION DES FICHIERS DSMS *
:/ ADD NAME=$RADP.DINS
//* DSMS : LISTE DES MODULES INSTALLES *
:/ ADD NAME=$RADP.DLDE
//* DSMS : CHARGEMENT BASE DES LIBELLES D'ERREURS 'DE' *
:/ ADD NAME=$RADP.DLDZ
//* DSMS : CHARGEMENT BASE DES SPA 'DZ' *
:/ ADD NAME=$RADP.DLVB
//* DSMS : REMPLACEMENT DES LOW-VALUES EN BLANCS *
:/ ADD NAME=$RADP.DPRT
//* DSMS : EDITIONS ET REQUETES *
:/ ADD NAME=$RADP.DREN
//* DSMS : CHANGEMENT CODES TABLE, MOTS CLES ET SITES *
:/ ADD NAME=$RADP.DREO
//* DSMS : REORGANISATION DSMS *
:/ ADD NAME=$RADP.DRST
//* DSMS : RECHARGEMENT-RESTAURATION DE LA BASE DSMS *
:/ ADD NAME=$RADP.DR8Q
//* DSMS : REPRISE DE BASE DSMS 8.0.2 01 OU 02 *
:/ ADD NAME=$RADP.DR80
//* DSMS : REPRISE DE LA SAUVEGARDE DSMS 8.0.1 *
:/ ADD NAME=$RADP.DSAV
//* DSMS : SAUVEGARDE DE LA BASE DSMS *
:/ ADD NAME=$RADP.DUPT
//* DSMS : MISE A JOUR DE LA BASE DSMS *
:/ ADD NAME=$RADP.DUPD
//* DSMS : MISE A JOUR DE LA BASE DSMS - DAF *
:/ ADD NAME=$RADP.DXBJ
//* DSMS : EXTRACTION JOURNAL DSMS POUR MAJ DSMS *
:/ ADD NAME=$RADP.DR15
//* DSMS : REPRISE DE LA BASE DSMS 1.2 OU 1.5 *
:/ ADD NAME=$RADP.DR5J
//* DSMS : REPRISE DU JOURNAL SEQUENTIEL 1.2 OU 1.5 *
:/ ADD NAME=$RADP.DEXQ
//* DSMS : EXTRACTION JOURNAL DSMS - MAJ DSMS *
:/ ADD NAME=$RADP.LDDF
//* DSMS : CHARGEMENT BASE 'DF' (D.A.F.) *
:/ ADD NAME=$RADP.DPDF
//* DSMS : ACCESS FACILITY PREPROCESSING *

```

	PAGE	182
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
CHARGEMENT DE LA DBDLIB		4

19.5.4. CHARGEMENT DE LA DBDLIB

4. CHARGEMENT DE LA DBDLIB

La bande d'implantation contient les DBD des bases sous forme d'objets et sous forme de sources.

Le JOB '\$PRFJ.DBD' effectuée par IEBCOPY le chargement des DBD objets dans la DBDLIB.

Ce chargement n'est à exécuter que si la valeur choisie, dans la génération du JCL, pour :

.le suffixe des DBD (paramètre SUF) est '22'

.le type des disques supports des bases (paramètres UNITU et UNITV) est 3390,

et si la taille des CI (4096) fournie dans les paramètres des DEFINE des bases a été conservée.

Dans le cas contraire, une recompilation des sources doit être effectuée. Il faut alors en vérifier le contenu et, en particulier, si la taille des CI a été changée, adapter les macros SIZE et RMNAME.

Ces sources ont été catalogués dans le fichier des paramètres SY, sous le nom PACDxxyy avec pour yy la valeur du paramètre SUF et pour xx les valeurs DA, D1, DE, DJ, DX, DZ, DF et F1.

NOTE : Utiliser la procédure de compilation standard d'IBM.

(voir JCL ci-après.)

INSTALLATION
DEROULEMENT DE L'INSTALLATION
CHARGEMENT DE LA DBDLIB

19
5
4

```
//SPRFJ.DBD JOB ($CCPT), 'DBDLIB', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
//ET010 EXEC PGM=IEBCOPY  
//*****  
//* CHARGEMENT DE LA DBDLIB - DSMS *  
//*****  
//*  
//*-----*  
//* A N'EXECUTER QUE SI VOTRE TYPE DE DISQUE EST '3390' ET *  
//* QUE LE SUFFIXE DES DBD EST '22'. SINON, IL FAUT COMPILER *  
//* LES SOURCES DES DBD PRESENTS DANS LE FICHER PDS DES *  
//* PARAMETRES 'SY'. *  
//*-----*  
//*  
//SYSPRINT DD SYSOUT=$OUT  
//SYSUT3 DD UNIT=$UWK, SPACE=(TRK, 5)  
//SYSUT4 DD UNIT=$UWK, SPACE=(TRK, 5)  
//IND1 DD DSN=PACD.DBDLIB, DISP=OLD, UNIT=$UTAPE, LABEL=(06, SL),  
// VOL=(, RETAIN, SER=$TAPEI)  
//OUT1 DD DSN=$DBDLIB, DISP=SHR  
//SYSIN DD *  
COPY INDD=((IND1,R)), OUTDD=OUT1  
S M=PACDDA22  
S M=PACDDE22  
S M=PACDDJ22  
S M=PACDDX22  
S M=PACDDZ22  
S M=PACDDF22  
S M=PACDD122  
S M=PACDF122  
/*  
/*  
/*
```

INSTALLATION	
DEROULEMENT DE L'INSTALLATION	
CHARGEMENT DE LA PSBLIB	

19
5
5

19.5.5. CHARGEMENT DE LA PSBLIB

5. CHARGEMENT DE LA PSBLIB

La bande d'implantation contient les PSB sous forme d'objets et sous forme de sources.

Le JOB '\$PRFJ.PSB' effectuée par IEBCOPY le chargement des PSB objets dans la PSBLIB.

Ce chargement n'est à exécuter que si la valeur choisie, dans la génération du JCL, pour :

```
.le suffixe des DBDs (paramètre SUF) est      : '22'
.le radical du système (paramètre ROOT) est   : 'P2'
.le suffixe des PSBs batch (paramètre SUG) est: '22'
```

Dans le cas contraire, une recompilation des sources doit être effectuée. Ces sources ont été catalogués dans le fichier des paramètres SY, sous le nom XXXXXXzz pour les PSB batch, rrYYYY pour les PSB TP, avec pour zz la valeur du paramètre SUG, pour rr la valeur du paramètre ROOT.

Pour XXXXXX, les valeurs sont:

```
PDSB   PDSBAS PDSEX  PDSR10 PDSR40 PDSUP0 PDSV10 PDSV20
PDSXDT PDS300 PDS320 PDS380 PDS400 PDS450 PDS500 PDS600
PDS610 PDS700 PDSINI PDSXTH PDSMS  PDSDAF PTV090
```

Pour YYYY, les valeurs sont:

00ZZ et 0099

NOTE : Utiliser la procédure de compilation standard d'IBM.

(voir JCL ci-après.)

INSTALLATION
 DEROULEMENT DE L'INSTALLATION
 CHARGEMENT DE LA PSBLIB

19
 5
 5

```
//$PRFJ.PSB JOB ($CCPT),'PSBLIB',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//ET010 EXEC PGM=IEBCOPY
//*****
//*          CHARGEMENT DE LA PSBLIB          - DSMS *
//*****
//*
//*-----*
//* A N'EXECUTER QUE SI: *
//* - LE RADICAL DU SYSTEME CHOISI EST:  ROOT='P2' *
//* - LE SUFFIXE DES BASES EST .....:  SUF='22' *
//* - LE SUFFIXE DES PSB BATCH EST ...:  SUG='22' *
//* SI UNE SEULE DE CES 3 CONDITIONS N'EST PAS REALISEE, IL *
//* FAUT COMPILER LES SOURCES PSB PRESENTS DANS LE FICHIER *
//* PDS DES PARAMETRES 'SY'. *
//*-----*
//*
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3 DD UNIT=$UWK,SPACE=(TRK,15)
//SYSUT4 DD UNIT=$UWK,SPACE=(TRK,15)
//IND1 DD DSN=PACD.PSBLIB,DISP=OLD,UNIT=$UTAPE,LABEL=(07,SL),
// VOL=(,RETAIN,SER=$TAPEI)
//OUT1 DD DSN=$PSBLIB,DISP=SHR
//SYSIN DD *
COPY INDD=((IND1,R)),OUTDD=OUT1
S M=PDSB22
S M=PDSBAS22
S M=PDSEX22
S M=PDSINI22
S M=PDSR1022
S M=PDSR4022
S M=PDSUP022
S M=PDSV1022
S M=PDSV2022
S M=PDSXDT22
S M=PDSXTH22
S M=PDS30022
S M=PDS32022
S M=PDS38022
S M=PDS40022
S M=PDS45022
S M=PDS50022
S M=PDS60022
S M=PDS61022
S M=PDS70022
===SEQ FOR DAF
S M=PDSDAF22
===SEQ
S M=PTV09022
S M=PDSMS22
S M=P200ZZ
S M=P20099
/*
/**
/**
```

INSTALLATION	PAGE	186
DEROULEMENT DE L'INSTALLATION		19
PREPARATION INITIALE DES FICHIERS		5
		6

19.5.6. PREPARATION INITIALE DES FICHIERS

6. PREPARATION INITIALE DES FICHIERS

Cette préparation, qui doit être exécutée uniquement lors de la première implantation, est constituée d'un job '\$PRFJ.PRE' qui comprend les étapes suivantes:

```
ET010 : IEHPROGM : SCRATCH/UNCATLG des DSCB modèles
ET020 : IEFBR14  : allocation des DSCB
ET030 : IDCAMS   : BLDG index data-group et
                  initialisation du fichier 'BB'
                  (sauvegarde de la base DSMS)
ET040 : IDCAMS   : BLDG index data-group et
                  initialisation du fichier 'BJ'
                  (JOURNAL archivé)
ET050 : IDCAMS   : chargement du fichier 'BB'
ET060 : IEBGENER : chargement du fichier 'BJ'
ET070 : $RADP.DLDZ : chargement de la base des SPA 'DZ'
ET075 : $RADP.LDDF : chargement du fichier de travail 'DF'
```

NOTE : Utiliser la procédure de compilation standard d'IBM.

(voir JCL ci-après.)

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

PREPARATION INITIALE DES FICHIERS

6

```

//$PRFJ.PRE JOB ($CCPT), 'PREPARATION', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//*   JOB NE DEVANT PASSER QU'A LA PREMIERE IMPLANTATION DSMS   *
//*****
// JCLLIB ORDER=( $BIBP )
//ET010 EXEC PGM=IEHPRGM
//*****
//*   SRATCH-UNCATLG DSCB   *
//*****
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD UNIT=$UNITP, VOL=SER=$VOLP, DISP=SHR
//SYSIN DD *
        UNCATLG DSN=$INDEXQ..DSCB.$ROOT.$FILE.BB
        SCRATCH DSN=$INDEXQ..DSCB.$ROOT.$FILE.BB, VOL=$UNITP=$VOLP
        UNCATLG DSN=$INDEXQ..DSCB.$ROOT.$FILE.BJ
        SCRATCH DSN=$INDEXQ..DSCB.$ROOT.$FILE.BJ, VOL=$UNITP=$VOLP
/*
//ET020 EXEC PGM=IEFBR14
//DDA DD DSN=$INDEXQ..DSCB.$ROOT.$FILE.BB, DISP=( ,CATLG,DELETE),
//      SPACE=(TRK,(0)), VOL=SER=$VOLP, UNIT=$UNITP,
//      DCB=(RECFM=VB, LRECL=354, BLKSIZE=6376)
//DDB DD DSN=$INDEXQ..DSCB.$ROOT.$FILE.BJ, DISP=( ,CATLG,DELETE),
//      SPACE=(TRK,(0)), VOL=SER=$VOLP, UNIT=$UNITP,
//      DCB=(RECFM=FB, LRECL=180, BLKSIZE=6300)
//*
//*****
//*   CONSTRUCTION INDEX FICHIERS A GENERATION   *
//*****
//ET030 EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$CATV, DISP=SHR
//*: DD DSN=$CATU, DISP=SHR
//DD1 DD DSN=$INDEXQ..$ROOT.$FILE.BB, DISP=( ,CATLG,DELETE),
//      UNIT=$UNITO, VOL=SER=$VOLO, SPACE=(TRK,0),
//      DCB=$INDEXQ..DSCB.$ROOT.$FILE.BB
//SYSIN DD *
        DEFINE GENERATIONDATAGROUP -
            (NAME ($INDEXQ..$ROOT.$FILE.BB) LIMIT (3) SCR)
//SYSPRINT DD SYSOUT=$OUT
//*
//ET040 EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$CATV, DISP=SHR
//*: DD DSN=$CATU, DISP=SHR
//DD1 DD DSN=$INDEXQ..$ROOT.$FILE.BJ, DISP=( ,CATLG,DELETE),
//      UNIT=$UNITO, VOL=SER=$VOLO, SPACE=(TRK,0),
//      DCB=$INDEXQ..DSCB.$ROOT.$FILE.BJ
//SYSIN DD *
        DEFINE GENERATIONDATAGROUP -
            (NAME ($INDEXQ..$ROOT.$FILE.BJ) LIMIT (3) SCR)
//SYSPRINT DD SYSOUT=$OUT
//*
//ET050 EXEC PGM=IDCAMS
//*****
//*   CHARGEMENT DE BB   *
//*****
//*:STEPCAT DD DSN=$CATV, DISP=SHR
//*: DD DSN=$CATU, DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD DSN=PAC.BB, DISP=OLD, UNIT=$UTAPE, LABEL=(08,SL),
//      VOL=( ,RETAIN, SER=$TAPEI)
//DD2 DD DSN=$INDEXQ..$ROOT.$FILE.BB(+1), DISP=( ,CATLG,DELETE),
//      VOL=SER=$VOLO, SPACE=(TRK,(100,10),RLSE), UNIT=$UNITO,
//      DCB=$INDEXQ..DSCB.$ROOT.$FILE.BB
//SYSIN DD *
        REPRO INFILE(DD1) OUTFILE(DD2)
//ET060 EXEC PGM=IEBGENER
//*****
//*   CHARGEMENT DE BJ   *
//*****
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT1 DD DUMMY, DCB=$INDEXQ..DSCB.$ROOT.$FILE.BJ
//SYSUT2 DD DSN=$INDEXQ..$ROOT.$FILE.BJ(+1), DISP=( ,CATLG,DELETE),

```

INSTALLATION
DEROULEMENT DE L'INSTALLATION
PREPARATION INITIALE DES FICHIERS

19
5
6

```
//          VOL=SER=$VOLO,SPACE=(TRK,(1,1),RLSE),UNIT=$UNITO,  
//          DCB=$INDEXQ..DSCB.$ROOT.$FILE.BJ  
//*  
//*****  
//*          INITIALISATION DE DZ          *  
//*****  
//ET070    EXEC $RADP.DLDZ  
//*  
//*****  
//*          INITIALISATION DE DF          *  
//*****  
//ET075    EXEC $RADP.LDDF  
//*  
//
```

INSTALLATION	PAGE	189
DEROULEMENT DE L'INSTALLATION		19
CHARGEMENT DES LIBELLES D'ERREUR		5
		7

19.5.7. CHARGEMENT DES LIBELLES D'ERREUR

7. CHARGEMENT DES LIBELLES D'ERREUR DSMS

Ce chargement crée la base 'DE' des libellés d'erreur du système. Il est constitué d'un JOB '\$PRFJ.DE' qui comprend une seule étape :

ET010 : \$RADP.DLDE : chargement de DE à partir du fichier
séquentiel fourni sur la bande.

(voir JCL ci-après.)

INSTALLATION
DEROULEMENT DE L'INSTALLATION
CHARGEMENT DES LIBELLES D'ERREUR

19
5
7

```
//$PRFJ.DE JOB ($CCPT),'LOADING -DE-',CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
//*****  
//* CHARGEMENT DE LA BASE LIBELLES D'ERREURS DU MODULE DSMS *  
//*****  
// JCLLIB ORDER=( $BIBP )  
// *  
//ET010 EXEC $RADP.DLDE  
//PDSV10.PAC7IN DD DSN=PAC.DE,DISP=OLD,UNIT=$UTAPE,LABEL=(09,SL),  
// VOL=( ,RETAIN,SER=$TAPEI )  
// *  
//
```

	PAGE	191
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
CHARGEMENT DES ACB		8

19.5.8. CHARGEMENT DES ACB

8. COMPILATION DES ACB

Cette étape consiste à construire tous les ACB qui permettront d'utiliser le système DSMS en TP.

NOTE : Utiliser la procédure de compilation standard d'IBM.

Important

La DBDLIB doit être concaténée avec la DBDLIB du système VA Pac (voir le paramètre \$DBDLIO).

(Voir JCL ci-après.)

INSTALLATION
DEROULEMENT DE L'INSTALLATION
CHARGEMENT DES ACB

19
5
8

```
//$PRFJ.ACB JOB ($CCPT), 'ACBGEN', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
//ET010 EXEC ACBGEN, SOUT=$OUT, PSB='$PSBLIB',  
// DBD='$DBDLIB', ACB='$ACBLIB'  
//*****  
//* COMPILATION DES ACB - DSMS *  
//*****  
//*  
//*-----*  
//* CE JOB DOIT ETRE EXECUTE APRES LE CHARGEMENT DES DBDLIB *  
//* ET PSBLIB (SOIT PAR MODULE OBJETS, SOIT PAR COMPILATION *  
//* DES SOURCES PSB ET DBD). *  
//* *  
//* IMPORTANT : LA DBDLIB DSMS DOIT ETRE CONCATENEE *  
//* AVEC LA DBDLIB DE VISUALAGE PACBASE *  
//*-----*  
//*  
//G.SYSIN DD *  
BUILD DBD=(PACDDA$SUF, PACDDC$SUF, PACDDE$SUF, PACDDJ$SUF)  
BUILD DBD=(PACDDX$SUF, PACDDZ$SUF, PACDDF$SUF, PACDD1$SUF)  
BUILD DBD=(PACDD3$SUF, PACDF1$SUF)  
BUILD PSB=($ROOT.00ZZ, $ROOT.0099)  
/*  
/*  
/*
```


INSTALLATION	PAGE	193
DEROULEMENT DE L'INSTALLATION		19
COMPLEMENT : SOURCES POINT ENTREE UTILISATEUR		5
		9

19.5.9. COMPLEMENT : SOURCES POINT ENTREE UTILISATEUR

9. COMPLEMENT D'INSTALLATION

Ce complément d'installation n'est à effectuer que pour les utilisateurs désirant ajouter des contrôles sur la saisie des fiches amélioration, événement, requête, maquette ou site et pour les utilisateurs de DAF.

Job '\$prfj.7SRC'

STEP1 : IDCAMS : DELETE du PDS.
 STEP2 : IEFBR14 : allocation du PDS.
 STEP3 : IEBCOPY : chargement des membres du PDS.

9.1 LES SOURCES POUR CONTROLES UTILISATEUR

Cinq sous-programmes batch et cinq TP sont livrés, afin que l'administrateur de DSMS puisse y insérer des contrôles supplémentaires. Ils doivent être compilés et "linkés" dans les bibliothèques de load-modules DSMS.

Pour les rendre actifs, il faut le préciser lors de la restauration (Voir le chapitre consacré à la procédure DRST).

LES MOUVEMENTS BATCH POUR DAF

Ces mouvements pouvant être utilisés pour l'écriture de programmes utilisant le module DAF, ils doivent être mis à jour dans la base VA Pac par la procédure UPDT. (Voir le sous-chapitre suivant "Installation de l'environnement DAF").

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

COMPLEMENT : SOURCES POINT ENTREE UTILISATEUR

9

```

//$PRFJ.SRC JOB ($CCPT), 'SOURCES', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//* DSMS *
// * *
// * - INSTALLATION - PACSSRC- *
// * *
//*****
// *
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN DD *
DELETE ($INDEXP..$ROOT.$ROOT.SRC)
// *
//STEP2 EXEC PGM=IEFBR14
//SY DD DSN=$INDEXP..$ROOT.$ROOT.SRC,DISP=(,CATLG,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
// UNIT=$UNITP,
// VOL=SER=$VOLP,
// SPACE=(6080,(50,10,10))
// *
//STEP3 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3 DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTM DD DSN=$INDEXP..$ROOT.$ROOT.SRC,DISP=OLD
//INM DD DSN=PACD.SOURCE,DISP=SHR,
// VOL=(,RETAIN,SER=$TAPEI),UNIT=$UTAPE,LABEL=(10,SL)
//SYSIN DD *
C I=INM,O=OUTM
S M=PDSCAM
S M=PDSCFV
S M=PDSCMQ
S M=PDSCRQ
S M=PDSCSI
S M=((PWCUMQ,$ROOT.CUMQ))
S M=((PWCUEV,$ROOT.CUEV))
S M=((PWCUMQ,$ROOT.CUMQ))
S M=((PWCURQ,$ROOT.CURQ))
S M=((PWCUSI,$ROOT.CUSI))
S M=((DAFDICF,DAFDIC))
//

```

	PAGE	195
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
COMPLEMENT : SOURCES POINT ENTREE UTILISATEUR		9

9.2 INSTALLATION ENVIRONNEMENT DAF

Le principe de DAF implique la transformation des requêtes SQL d'accès à la base DSMS écrites dans les programmes utilisateur, par la génération de données et d'appels de sous-programmes d'accès à la base dans le source Cobol généré de ces programmes.

Le pré-processeur traite donc les programmes générés pour effectuer cette transformation.

Le pré-processeur comporte le programme DAFD10 installé dans la bibliothèque des load-modules batch.

L'utilisateur doit utiliser la procédure DPDF pour traiter ses programmes générés utilisant DAF :

- . par l'appel de cette procédure dans les cartes de contrôle optionnelles avant/après programme, combinées au JCL de compilation-link-edit,
- . par l'appel de cette procédure derrière l'exécution de la procédure GPRT standard d'où sera repris le flot généré,
- . par toute autre méthode la plus adaptée aux contraintes de l'exploitation sur le site.

(voir le sous-chapitre consacré à DPDF).

	PAGE	196
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
COMPLEMENT : SOURCES POINT ENTREE UTILISATEUR		9

Un SOUS-PROGRAMME DAF est fourni à l'installation :

PDTPDF pour les requêtes DAF standard,
 REMARQUE : Ce sous-programme doit être transféré dans la ou les bibliothèques de programmes utilisateur, soit pour être inclus au "Link-Edit" de ces programmes (appel statique), soit pour être appelés à l'exécution (appel dynamique).

Un EXEMPLE DE JCL de programme batch utilisateur appelant DAF est fourni dans le PDS des JCL dans le membre PACSDAF (Voir ci-dessous un exemple de JCL DAF batch utilisateur).

Le fichier de travail nécessaire au fonctionnement de DAF TP a un DDNAME imposé de la forme \$root.\$root.DF. Il doit être unique pour l'ensemble des programmes accédant à la même base DSMS. Les ENTITES RUBRIQUES, STRUCTURES DE DONNEES, SEGMENTS, pouvant être utilisées pour l'écriture de programmes utilisant le module DAF, sont fournies sous forme de mouvements batch dans le membre DAFDIC du PDS des Compléments SRC.

IMPORTANT :

L'introduction de ce "dictionnaire DAF" dans la base VA Pac par la procédure de mise à jour batch UPDT est sous la responsabilité du gestionnaire de la base qui doit AU PREALABLE S'ASSURER DE LA COMPATIBILITE DES CODES ENTITES FOURNIS avec les entités déjà existantes dans son réseau.

Afin d'éviter les problèmes de compatibilité du dictionnaire du site avec les entités fournies pour le module DAF, il est conseillé de créer un réseau indépendant de bibliothèques pour l'écriture des utilitaires DAF du site.

Il est cependant possible de l'insérer dans la même bibliothèque que le dictionnaire PAF.

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

COMPLEMENT : SOURCES POINT ENTREE UTILISATEUR

9

```

//*****
//* DSMS 2.5
//* - EXEMPLE DE JCL -
//* EXECUTION D'UN PROGRAMME UTILISATEUR D.A.F. BATCH
//*****
//DAFBATCH PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEX='$INDEX', INDEX VSAM
//*: SYSTCAT='$CATV', CATALOGUE VSAM SYSTEME
//*: VSAMCAT='$CATU', CATALOGUE VSAM UTILISATEUR
// OUT='$OUT', CLASSE DE SORTIE
// INDEXP='$INDEXP', INDEX FICHIERS NON VSAM
// STEPLIB='$MODB', BIBLI. DES LOAD-MODULES BATCH
// PSBLIB='$PSBLIB', BIBLI. DES PSB
// DBDLIB='$DBDLIB', BIBLI. DES DBD
// DBDLIO='$DBDLIO', BIBLI. DES DBD VA PAC
// RESLIB='$RESLIB', RESLIB IMS
// PROCLIB='$PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//WITHDAF EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,_____,_____$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
// COND=(00,NE,EXISDJ)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7DA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PAC7DE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DF DD DSN=&INDEX..&ROOT.&ROOT.DF,DISP=SHR
//PAC7F1 DD DSN=&INDEX..&ROOT.&ROOT.F1,DISP=SHR
//----- DD DSN=---
//----- DD DSN=---
//----- DD DSN=---
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
// PEND
//DAFBATCH EXEC DAFBATCH

```

	PAGE	198
INSTALLATION		19
DEROULEMENT DE L'INSTALLATION		5
CHARGEMENT DE LA BASE DE TEST DSMS		10

19.5.10. CHARGEMENT DE LA BASE DE TEST DSMS

10. CHARGEMENT DE LA BASE DSMS DE TESTS

Avant d'effectuer les tests, l'utilisateur doit charger la base DSMS de tests par l'exécution du job '\$prfj.DS'.

Ce job exécute la procédure DRST avec en entrée la sauvegarde qui a été chargée sur disque à l'étape ET050 du job '\$prfj.PRE' (Voir paragraphe 6 ci-dessus).

(Voir JCL ci-après.)

INSTALLATION

19

DEROULEMENT DE L'INSTALLATION

5

CHARGEMENT DE LA BASE DE TEST DSMS

10

```
//$PRFJ.DS JOB ($CCPT),'TESTS',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*
//*****
//*          RESTAURATION FICHIERS TEST DU MODULE DSMS          *
//*****
//*
//*-----*
//* UNE FOIS CETTE RESTAURATION CORRECTEMENT EXECUTEE, LA NOU-*
//* VELLE BASE DSMS EST INSTALLEE. LA PREMIERE CONNEXION S'EF-*
//* FECTUE DE LA MANIERE SUIVANTE:                               *
//*   - ENTRER LE CODE TRANSACTION DSMS                         *
//*   - ENTRER SUR L'ECRAN INITIAL,                              *
//*     LE CODE UTILISATEUR: '*****'                          *
//*     LE MOT DE PASSE...: '*****'                            *
//*     LE CHOIX.....: 'HT'                                     *
//* (POUR LES CHOIX AUTORISES DANS CE CAS PRECIS, SE REPORTER *
//* AUX EXPLICATIONS DONNEES DANS LA PROCEDURE 'DINI' DECRITE *
//* DANS LE MANUEL D'EXPLOITATION).                             *
//*-----*
// JCLLIB ORDER=( $BIBP)
//STEP01 EXEC $RADP.DRST
RF
//*
//
```

19.6. LISTE DES PROGRAMMES INSTALLEES

11. LISTE DES PROGRAMMES INSTALLEES :

Cette liste s'obtient par le job '\$PRFJ.INSL' qui exécute la procédure '\$RADP.DINS' ; elle contient :

- . La liste des programmes Batch et TP avec les dates de compilation.

Cette liste doit être conservée pour permettre de transmettre à IBM les références de l'installation en cas de détection d'anomalies dans le fonctionnement du système DSMS.

(Voir JCL ci-après.)

INSTALLATION
LISTE DES PROGRAMMES INSTALLES

19
6

```
//$PRFJ.INSL JOB ($CCPT), 'INSTALLATION LIST', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// *  
//*****  
// * CE JOB FOURNI LES LISTES DES PROGRAMMES INSTALLES SUR LE SITE *  
//*****  
// *  
// JCLLIB ORDER=($BIBP)  
//STEP01 EXEC $RADP.DINS  
$ROOT  
// *  
//
```

INSTALLATION
LISTE DES PROGRAMMES INSTALLES

19
6

```

//*****
//* DSMS 2.5
//*
//* LISTE DES MODULES INSTALLES
//*****
//$RADP.DINS PROC FILE=$FILE, NUMERO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// INDEXP=' $INDEXP', INDEX FICHIERS NON VSAM
// INDEX=' $INDEX', INDEX FICHIERS VSAM
//*: VSAMCAT=' $CATU', CATALOGUE VSAM UTILISATEUR
// OUT=' $OUT', CLASSE DE SORTIE
// MODB=' $MODB', BIBLIOTHEQUE LM BATCH INSTALLES
// MODT=' $MODT', BIBLIOTHEQUE LM TP INSTALLES
// PSBLIB=' $PSBLIB', BIBLIOTHEQUE DES PSB
// DBDLIB=' $DBDLIB', BIBLIOTHEQUE DES DBD
// RESLIB=' $RESLIB', RESLIB IMS
// PROCLIB=' $PRCLIB', PROCLIB IMS
// UWK=$UWK, UNITE DE TRAVAIL
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* ENTREES :
//* - 1ERE LIGNE : RADICAL DU SYSTEME DSMS (COL.3 LONGUEUR 2)
//* - AUTRES LIGNES SI UNE SELECTION DES PROGRAMMES EST NECESSAIRE:
//* UNE LIGNE PAR PROGRAMME: CODE PROGRAMME (COL.3 LONGUEUR 6)
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&MODB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PAC7MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
//*
//PDSXDT EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSXDT,PDSXDT$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&MODB,DISP=SHR
// DD DSN=&MODT,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDDS DD SYSOUT=&OUT
//PACDMB DD DSN=&&PAC7MB,DISP=(OLD,PASS)
//*

```

19.7. TESTS D'UTILISATION

TESTS D'UTILISATION

TESTS DE L'INSTALLATION DE DSMS

Ces tests comprennent trois phases :

- . Tests d'utilisation de la Base en TP,
- . Test de l'utilitaire d'extraction,
- . Tests de gestion de la Base.

1. TESTS D'UTILISATION TP

Ouvrir les fichiers de la Base de test sous le TP.
Se connecter avec l'utilisateur TEST et le mot de passe IBM.
Effectuer des enchaînements d'écrans et des mises à jour.

2. TEST D'EXTRACTION

Exécuter le job '\$prfjDEXT' (Procédure DEXT) qui
extrait des éléments de la base de tests.
Pour ce test, les fichiers de la Base peuvent rester ouverts en TP.

3. TESTS DE GESTION DE LA BASE

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 '\$prfjDARC' donnant un fichier BJ(1).
- . Sauvegarde directe de la base : exécuter le job '\$prfjDSAV' donnant un fichier BB(1).
- . Restauration de la base à partir de l'archive BJ(1) et de la sauvegarde de la base BB(1) : exécuter le job '\$prfjDRST'.

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

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

INSTALLATION
 TESTS D'UTILISATION
 JCL TEST : DEXT

19
 7
 1

19.7.1. JCL TEST : DEXT

```
//$PRFJ.DXT JOB ($CCPT), 'EXTRACTION', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=( $BIBP)
//*****
// * TESTS D'EXTRACTION : EXTRACTION DES TABLES UTILISATEURS *
// * EXTRACTION D'UNE REQUETE *
//*****
//DEXT EXEC $RADP.DEXT
 *TEST IBM
TUD
TUG
TUS
TUP
QC LISTE
QE LCHECK $/19931010/
//PDSEX.PACDIM DD SYSOUT=$OUT
//*
//* PDSEX.PACDIM DD DSN=&&PACDIM, DISP=( ,PASS),
//* UNIT=&UWK, SPACE=(TRK,(1,1),RLSE),
//* DCB=(RECFM=FB,LRECL=250,BLKSIZE=5000)
//* DUPT EXEC $RADP.DUPT
//* PDSUP0.PACDIM DD DSN=&&PACDIM, DISP=(OLD,PASS),
//* DCB=BLKSIZE=5000
//
```

INSTALLATION
TESTS D'UTILISATION
JCL TEST : DARC

19
7
2

19.7.2. JCL TEST : DARC

```
//$PRFJ.DAR JOB ($CCPT), 'DARC', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// JCLLIB ORDER=( $BIBP )  
//*****  
//*          TEST D'ARCHIVAGE DU JOURNAL          *  
//*****  
//DARC EXEC $RADP.DARC
```

INSTALLATION
TESTS D'UTILISATION
JCL TEST : DSAV

19
7
3

19.7.3. JCL TEST : DSAV

```
//$PRFJ.DSA JOB ($CCPT), 'DSAV', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// JCLLIB ORDER=( $BIBP )  
//*****  
//*          TEST DE LA SAUVEGARDE DE LA BASE          *  
//*****  
//DSAV EXEC $RADP.DSAV
```

INSTALLATION
TESTS D'UTILISATION
JCL TEST : DRST

19
7
4

19.7.4. JCL TEST : DRST

```
//$PRFJ.DRS JOB ($CCPT), 'DRST', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// JCLLIB ORDER=( $BIBP )  
//*****  
//* TESTS DE RESTAURATION DE LA BASE DSMS AVEC LE JOURNAL *  
//* *  
//* APRES LA RESTAURATION, REFAIRE QUELQUES TESTS RAPIDES DE *  
//* FONCTIONNEMENT EN TP, APRES AVOIR REOUVERT LES FICHIERS *  
//*****  
//DRST EXEC $RADP.DRST  
RF REC  
//*
```


INSTALLATION
TESTS D'UTILISATION
JCL TEST : DXBJ

19
7
5

19.7.5. JCL TEST : DXBJ

```
//$PRFJ.DXB JOB ($CCPT), 'DXBJ', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// JCLLIB ORDER=( $BIBP )  
//*****  
// * TESTS D'EXTRACTION DU JOURNAL DSMS *  
// * SELECTION SUR DATE, HEURE ET UTILISATEUR *  
//*****  
//DXBJ EXEC $RADP.DXBJ  
 *TEST IBM  
K 1994070119940715000000240000USER  
//PDS700.PACDIM DD SYSOUT=$OUT  
//*  
// *PDS700.PACDIM DD DSN=&&PACDIM, DISP=(, PASS),  
// * UNIT=&UWK, SPACE=(TRK, (15, 5), RLSE),  
// * DCB=(RECFM=FB, LRECL=250, BLKSIZE=5000)  
// *DUPT EXEC $RADP.DUPT  
// *PDSUP0.PACDIM DD DSN=&&PACDIM, DISP=(OLD, PASS),  
// * DCB=BLKSIZE=5000  
//  
//
```

19.8. REIMPLANTATION DU SYSTEME

REIMPLANTATION DU SYSTEME

Une réimplantation du système DSMS 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.

Trois cas peuvent se présenter :

Cas 1 : les JCLs d'implantation ont été conservés

Cas 2 : les JCLs d'implantation doivent être régénérés et la réimplantation est standard.

Cas 3 : les JCLs d'implantation doivent être régénérés et la réimplantation est non standard.

CAS 1 : VOUS AVEZ GARDE LES JCLs D'INSTALLATION

La REIMPLANTATION STANDARD consiste à exécuter les jobs contenus dans les modules de JCL suivants :

- PACSLOAD: job \$prfj.LOAD chargement des programmes.
- PACSDE : job \$prfj.DE chargement libellés erreurs.

1) PACSLOAD: rechargement des load-modules batch et TP
(Modifier le nom de la bande dans le paramètre VOL=SER=)

ATTENTION : le JOB fourni contient la suppression des bibliothèques de load-modules, leur allocation et la copie de tous les programmes. On peut donc procéder de deux manières:

- . soit par exécution du job complet : dans ce cas il est nécessaire de sauvegarder les programmes se trouvant dans les bibliothèques et qui ne proviennent pas directement de la bande d'installation (programmes utilisateur) ou qui ont fait l'objet d'une adaptation (cas des modules de reprise PDSR8B et PDSR8C) avant d'exécuter le job ;
- . soit par exécution de la seule étape de copie (IEBCOPY) des programmes (prévoir dans ce cas de supprimer les programmes au préalable afin de ne pas avoir de problème de place dans la bibliothèque).

2) PACSDE: IDCAMS DELETE/DEFINE et REPRO du fichier DE des libellés erreurs et documentation.
(Modifier le nom de la cartouche dans le paramètre VOL=SER=)

CAS 2 : VOUS DEVEZ REGENERER LES JCLs D'INSTALLATION POUR
UNE REIMPLANTATION STANDARD

Pour le détail des opérations à effectuer, voir dans le présent chapitre, les sous-chapitres "Préparation de l'installation" et "Installation du JCL complet".

La régénération des JCLs consiste à refaire une exécution de l'utilitaire MM1JCL avec les paramètres choisis lors de l'implantation sur le site, et en y ajoutant la sélection des JCLs utiles à la réimplantation.

Ajouter dans la SYSIN les lignes de sélection des Modules de JCL suivants :

```
===SELM PACSLOAD  
===SELM PACSDE
```

Vérifier les JCLs obtenus. Dérouler la réimplantation suivant les étapes décrites au CAS 1.

CAS 3 : VOUS DEVEZ REGENERER DES JCLs D'INSTALLATION DANS
LE CAS D'UNE REIMPLANTATION NON STANDARD

Reportez-vous au cas 2 pour l'obtention des JCL.

Une fois les JCLs obtenus, suivre les consignes particulières indiquées dans la note accompagnant l'envoi de la sous-version.

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
REPRISE DSMS 8.0.1 EN DSMS 2.5 (DR80)

PAGE 213

20

20. REPRISE DSMS 8.0.1 EN DSMS 2.5 (DR80)

20.1. OPERATIONS A EFFECTUER

OPERATIONS A EFFECTUER

La reprise de l'existant, qui permet le passage d'un réseau DSMS 8.0 ou 8.0.1 à un réseau DSMS exploitable par la nouvelle version, se compose des phases suivantes :

- . Archivage en version 8.0/8.0.1 de la base (Procédure DARC),
- . Sauvegarde en version 8.0/8.0.1 de la base, donnant un fichier BB 8.0/8.0.1 (Procédure DSAV).

Puis, avec la NOUVELLE INSTALLATION, exécution des procédures :

- . Conversion de la sauvegarde de la base DSMS 8.0/8.0.1 (BB) au nouveau format (procédure DR80),
- . Réorganisation de la sauvegarde dans le but de reconstruire les références croisées (fichier DX) au format de la nouvelle version (procédure DREO),
- . Restauration de la base (procédure DRST).

20.2. ENTREE UTILISATEUR

DR80 : ENTREE UTILISATEUR

L'entrée utilisateur permet la transposition des codes produit de 1 à 3 caractères.
Elle est composée de 1 à n lignes, chaque ligne étant décomposée en groupes de 4 caractères, à partir de la première colonne.

```
-----  
!Pos.!Lon.! Valeur ! Signification !  
!-----!  
! 1 ! 1 ! 'P' ! Ancien code produit !  
! 2 ! 3 ! 'PRO' ! Nouveau code produit !  
! ! ! ! Chaque groupe de 4 caractères peut !  
! ! ! ! être répété 20 fois maximum par ligne !  
-----
```

20.3. REPRISE DE LA SAUVEGARDE

DR80 : DESCRIPTION DES ETAPES

PRISE EN COMPTE DES ENTREES : PTU001

PREPARATION CONVERSION : PDSR8B

.Fichier permanent en entrée :
-Sauvegarde base DSMS 8.0.1
PACDBB : DSN=&OLDBB

.Fichiers de travail en sortie :

PACDIQ : DSN=&&PACDIQ
PACDIT : DSN=&&PACDIT
PACDIW : DSN=&&PACDIW
PACDLA : DSN=&&PACDLA

.Fichiers de tri :
SORTWK01, SORTWK02, SORTWK03

.Fichier mouvement en entrée :
PACDMB : DSN=&&DR80MB

.Etat en sortie :
-Compte rendu de reprise
PACDIK

CONVERSION SAUVEGARDE 8.0.2 : PDSR8C

.Fichiers de travail en entrée :
PACDIQ : DSN=&&PACDIQ
PACDIT : DSN=&&PACDIT
PACDIW : DSN=&&PACDIW
PACDLA : DSN=&&PACDLA

.Fichier permanent en entrée :
-Sauvegarde base DSMS 8.0.1
PACDBB : DSN=&OLDBB

.Fichier permanent en sortie :
-Sauvegarde base DSMS convertie
PACDB1 : DSN=&BBOLD

.Fichier mouvement en entrée :
PACDMB : DSN=&&DR80MB

.Fichiers de tri :
SORTWK01, SORTWK02, SORTWK03

.Etat en sortie :
-Compte rendu de reprise
PACDIO

CONVERSION SAUVEGARDE 1.2 : PDSR15

.Fichier permanent en entrée :
-Sauvegarde base DSMS 1.2 ou 1.5
PACDBB : DSN=&BBOLD

.Fichier permanent en sortie :

REPRISE DSMS 8.0.1 EN DSMS 2.5
REPRISE DE LA SAUVEGARDE

(DR80)

PAGE

217

20
3

-Sauvegarde base DSMS convertie
PACDB1 : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

.Etat en sortie :
-Compte-rendu d'édition
PACDIK

20.4. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5
//* - REPRISE DE BASE DSMS 8.0.1 -
//*****
//$RADP.DR80 PROC FILE=$FILE, NO DE BASE PHYSIQUE
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS
// OUT=' $OUT', CLASSE DE SORTIE
// INDEXQ=' $INDEXQ', INDEX FICHIERS A GENERATION
// CYL=' (4,1)', TAILLE WORKS DE TRI
// SPABB=' CYL, (10,5)', SPACE SAUVEGARDE
// SPAWK=' CYL, (10,5)', SPACE FICHIERS DE TRAVAIL
// UWK=$UWK, UNITE DE TRAVAIL
// OLDBB=, DSN SAUVEGARDE DE LA BASE 8.0.1
// VOLS='SER=$VOLO', VOLUME FICHER GENERATION
// UNITS=' $UNITO', UNITE FICHER GENERATION
// STEPLIB=' $MODB', BIBLIOTHEQUE LM BATCH
// SORTLIB=' $BIBT' BIBLIOTHEQUE DE TRI
//*-----*
//*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&DR80MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//PDSR8B EXEC PGM=PDSR8B
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDBB DD DSN=&OLDBB,DISP=OLD
//PACDMB DD DSN=&&DR80MB,DISP=(OLD,PASS)
//PACDLA DD DSN=&&PACDLA,DISP=(,PASS),
// UNIT=&UWK,SPACE=( &SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=1000)
//PACDIQ DD DSN=&&PACDIQ,DISP=(,PASS),
// UNIT=&UWK,SPACE=( &SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=284,BLKSIZE=5680)
//PACDIT DD DSN=&&PACDIT,DISP=(,PASS),
// UNIT=&UWK,SPACE=( &SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=104,BLKSIZE=2080)
//PACDIW DD DSN=&&PACDIW,DISP=(,PASS),
// UNIT=&UWK,SPACE=( &SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=92,BLKSIZE=2024)
//PACDIK DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PDSR8C EXEC PGM=PDSR8C
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDBB DD DSN=&OLDBB,DISP=OLD
//PACDMB DD DSN=&&DR80MB,DISP=(OLD,PASS)
//PACDLA DD DSN=&&PACDLA,DISP=(OLD,PASS)
//PACDIQ DD DSN=&&PACDIQ,DISP=(OLD,PASS)
//PACDIT DD DSN=&&PACDIT,DISP=(OLD,PASS)
//PACDIW DD DSN=&&PACDIW,DISP=(OLD,PASS)
//PACDB1 DD DSN=&&BBOLD,DISP=(,PASS),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=( &SPABB,RLSE),
// DCB=(RECFM=VB,BLKSIZE=6376,LRECL=354)

```

```
//PACDIO DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PDSR15 EXEC PGM=PDSR15
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PACDBB DD DSN=&&BBOLD,DISP=(OLD,PASS)
//PACDB1 DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,
//          SPACE=(&SPABB,RLSE),
//          DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB
//PACDIK DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
/*
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
REPRISE DSMS 8.0.2 01/02 EN DSMS 2.5 (DR8Q)

PAGE 220

21

21. REPRISE DSMS 8.0.2 01/02 EN DSMS 2.5 (DR8Q)

21.1. OPERATIONS A EFFECTUER

OPERATIONS A EFFECTUER

NOTE : ce chapitre ne concerne que les bases installées en 8.0.2 (version 01 ou 02); si une reprise DR8X ou DR80 vient d'être effectuée, ne pas procéder à cette nouvelle reprise.

L'installation de la nouvelle version de DSMS nécessite une reprise des requêtes dans la base DSMS, composée des étapes suivantes :

Avec l'installation 8.0.2 01 ou 02 :

1. Archivage de la base DSMS (DARC).
2. Sauvegarde de la base DSMS (DSAV).

Avec la NOUVELLE INSTALLATION :

3. Reprise du fichier BB issu de la sauvegarde (DR8Q).
4. Reorganisation de la sauvegarde (DREO).
5. Restauration de la base DSMS (DRST).

CONDITION D'EXECUTION

Aucune.

Cependant, pour des raisons de cohérence de la base reprise, il est conseillé de fermer l'accès au TP.

ENTREE UTILISATEUR

Aucune.

21.2. REPRISE DE LA SAUVEGARDE

DR8Q : DESCRIPTION DES ETAPES

REPRISE DES REQUETES : PDSR8Q

.Fichier permanent en entrée :
-Sauvegarde base DSMS 8.0.2 01/02
PACDBB : DSN=&OLDBB

.Fichier de travail en sortie :
PACDIQ : DSN=&&PACDIQ

.Fichiers de tri :
SORTWK01, 02, 03

FUSION : PDSR8R

.Fichier de travail en entrée :
PACDIQ : DSN=&&PACDIQ

.Fichier permanent en entrée :
-Sauvegarde base DSMS 8.0.2 01/02
PACDBB : DSN=&OLDBB

.Fichier permanent en sortie :
-Sauvegarde base DSMS reprise
PACDB1 : DSN=&BBOLD

CONVERSION SAUVEGARDE 1.2 : PDSR15

.Fichier permanent en entrée :
-Sauvegarde base DSMS 1.2 ou 1.5
PACDBB : DSN=&BBOLD

.Fichier permanent en sortie :
-Sauvegarde base DSMS convertie
PACDB1 : DSN=&INDEXQ.&ROOT.&FILE.BB(+1)

.Etat en sortie :
-Compte-rendu d'édition
PACDIK

21.3. JCL DE LA PROCEDURE

```

//*****
//* DSMS 2.5 *
//* REPRISE DE BASE DSMS 8.0.2 01 OU 02 EN 2.5 - *
//*****
//$RADP.DR8Q PROC ROOT='$ROOT', RADICAL DU SYSTEME DSMS
// FILE='$FILE', NUMERO DE BASE PHYSIQUE
// INDEXQ='$INDEXQ', INDEX FICHER A GENERATION
// STEPLIB='$MODB', BIBLIOTHEQUE LM BATCH
// OUT='$OUT', CLASSE DE SORTIE
// SORTLIB='$BIBT', BIBLIOTHEQUE DE TRI
// CYL=3, SPACE DES SORTWORK
// UWK=$UWK, UNITE TRAVAIL
// OLDBB=, DSNAME SAUVEGARDE DSMS 8.0.2 02
// VOLS='SER=$VOLO', VOLUME SAUVEGARDE
// UNITS='$UNITO', UNITE SAUVEGARDE (DISQUE OU BANDE)
// SPAWK=(CYL,(20,2)), SPACE FICHIERS DE TRAVAIL
// SPABB=(TRK,(45,5),RLSE) SPACE SAUVEGARDE
//*****
//PDSR8Q EXEC PGM=PDSR8Q,REGION=4096K
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//PACDBB DD DSN=&OLDBB,DISP=OLD
//PACDIQ DD DSN=&&PACDIQ,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAWK,
// DCB=(RECFM=FB,LRECL=284,BLKSIZE=5680)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PDSR8R EXEC PGM=PDSR8R,REGION=4096K
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACDBB DD DSN=&OLDBB,DISP=OLD
//PACDB1 DD DSN=&&BBOLD,
// DISP=(,PASS),
// UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
// DCB=(RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDIQ DD DSN=&&PACDIQ,DISP=(OLD,PASS)
//SYSUDUMP DD SYSOUT=&OUT
//*
//PDSR15 EXEC PGM=PDSR15
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PACDBB DD DSN=&&BBOLD,DISP=(OLD,PASS)
//PACDB1 DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=&SPABB,
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB
//PACDIK DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSOUPR DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*

```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
REPRISE DSMS 1.2 ou 1.5 en DSMS 2.5

PAGE 224

22

22. REPRISE DSMS 1.2 ou 1.5 en DSMS 2.5

22.1. PRESENTATION

OPERATIONS A EFFECTUER

La reprise de l'existant, qui permet le passage d'un réseau DSMS 1.2 ou 1.5 à un réseau DSMS exploitable par la nouvelle version, se compose des phases suivantes :

- . Archivage en version 1.2/1.5 de la base (Procédure DARC).
- . Sauvegarde en version 1.2/1.5 de la base, donnant un fichier BB 1.2/1.5 (Procédure DSAV).

Puis, avec la NOUVELLE INSTALLATION, exécution des procédures :

- . Conversion de la sauvegarde de la base DSMS 1.2/1.5 (BB) au nouveau format (procédure DR15).
- . Réorganisation de la sauvegarde dans le but de reconstruire les références croisées (fichier DX) au format de la nouvelle version (procédure DREO).
- . Restauration de la base (procédure DRST).

REMARQUE

Il est possible de reprendre la version séquentielle du journal 1.2 ou 1.5 par la procédure DR5J.

22.2. PROCEDURE 'DR15' - DESCRIPTION DES ETAPES

DR15 : DESCRIPTION DES ETAPES

REPRISE DES FICHIERS : PDSR15

.Fichier permanent en entrée :

-Sauvegarde base DSMS 1.2 ou 1.5
PACDBB : DSN=&OLDBB

.Fichier permanent en sortie :

-Sauvegarde base DSMS reprise
PACDB1 : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

.Etat en sortie:

-Compte-rendu d'édition
PACDIK

22.3. PROCEDURE 'DR15' : JCL

```
//*****  
//* DSMS 2.5 *  
//* - REPRISE DE BASE DSMS 1.2 OU 1.5 EN BASE DSMS 2.5 *  
//* *  
//*****  
//$RADP.DR15 PROC FILE=$FILE, NO DE BASE PHYSIQUE  
// ROOT=$ROOT, RADICAL DU SYSTEME DSMS  
// OUT='$OUT', CLASSE DE SORTIE  
// INDEXQ='$INDEXQ', INDEX FICHIERS A GENERATION  
// SPABB='(TRK,(45,5),RLSE)', SPACE SAUVEGARDE  
// OLDBB=, DSN SAUVEGARDE BASE 1.2 OU 1.5  
// VOLS='SER=$VOLO', VOLUME FICHER GENERATION  
// UNITS='$UNITE', UNITE FICHER GENERATION  
// STEPLIB='$MODB' BIBLIOTHEQUE LM BATCH  
//*-----*  
//*  
//PDSR15 EXEC PGM=PDSR15  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//PACDBB DD DSN=&OLDBB,DISP=OLD  
//PACDB1 DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),  
// UNIT=&UNITS,VOL=&VOLS,  
// SPACE=&SPABB,  
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB  
//PACDIK DD SYSOUT=&OUT  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//SYSPRINT DD SYSOUT=&OUT  
//SYSUDUMP DD SYSOUT=&OUT  
//*
```

22.4. PROCEDURE 'DR5J' - DESCRIPTION DES ETAPES

REPRISE DU JOURNAL: PDSR5J

.Fichier permanent en entrée :
-Journal séquentiel 1.2 ou 1.5
PACDBJ : DSN=&OLDBJ

.Fichier permanent en sortie :
-Journal repris au format 2.5
DSN=&INDEXQ..&ROOT.&FILE.BJ(+1)

.Etat en sortie:
-Compte-rendu d'édition
PACDIK

22.5. PROCEDURE 'DR5J' : JCL

```
//*****  
//* DSMS 2.5 *  
//* REPRISE DU JOURNAL SEQUENTIEL 1.2 OU 1.5 *  
//*****  
//$RADP.DR5J PROC FILE=$FILE, NO DE BASE PHYSIQUE  
// ROOT=$ROOT, RADICAL SYSTEME DSMS  
// OUT='$OUT', CLASSE DE SORTIE  
// INDEXQ='$INDEXQ', INDEX FICHIERS A GENERATION  
// SPABJ=(TRK,(45,5),RLSE)', SPACE SAUVEGARDE  
// OLDBJ=, DSN JOURNAL 1.2 OU 1.5  
// VOLS='SER=$VOLO', VOLUME FICHER GENERATION  
// UNITS='$UNITO', UNITE FICHER GENERATION  
// STEPLIB='$MODB' BIBLIOTHEQUE LM BATCH  
//*-----*  
//*  
//PDSR5J EXEC PGM=PDSR5J  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//PACDBJ DD DSN=&OLDBJ,DISP=OLD  
//PACDJB DD DSN=&INDEXQ..&ROOT.&FILE.BJ(+1),DISP=(,CATLG,DELETE),  
// UNIT=&UNITS,VOL=&VOLS,  
// SPACE=&SPABJ,  
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BJ  
//PACDIK DD SYSOUT=&OUT  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//SYSPRINT DD SYSOUT=&OUT  
//SYSUDUMP DD SYSOUT=&OUT  
//*
```

VisualAge Pacbase - Manuel d'Exploitation
DSMS - EXPLOITATION & INSTALLATION
REPLACEMENT DES LOW-VALUES PAR DES BLANCS (DLVB)

PAGE 230

23

23. REMPLACEMENT DES LOW-VALUES PAR DES BLANCS (DLVB)

23.1. DLVB : REPLACEMENT DES LOW-VALUES PAR DES BLANCS

REPLACEMENT DES "LOW VALUES" PAR DES BLANCS DANS UN FICHIER BB

La procédure DLVB se charge de remplacer par des blancs les "low-values" présentes dans le fichier de sauvegarde BB de la base.

L'objectif est de pouvoir transférer le fichier BB sur différentes plates-formes et d'éviter les problèmes liés à la présence de ces caractères lors des transferts.

Option d'utilisation :

La procédure DLVB donne à l'utilisateur la possibilité de ne garder en sortie que les enregistrements de type 'données'. Voir plus loin la mise en oeuvre de cette option.

Dans ce cas, la sauvegarde obtenue sur la plate-forme cible après transfert devra être réorganisée (procédure DREO) afin de reconstruire le fichier des références croisées (DX).

CONDITION D'EXECUTION

Aucune.

23.2. DLVB : PARAMETRES-DESCRIPTION DES ETAPES

DLVB : DESCRIPTION DES ETAPES

REPLACEMENT DES "LOW-VALUES" PAR DES BLANCS : PDSLVB

.Carte EXEC : indiquer PARM='DATA' pour ne garder dans le
fichier en sortie que les enregistrements DONNEES.
Ne rien indiquer pour conserver les enregistrements INDEX
et les enregistrements DONNEES.

.Fichier en entrée :
-Sauvegarde de la base
PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(0)

.Fichier en sortie :
-Sauvegarde de la base
PACDB1 : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

23.3. DLVB : PROCEDURE D'EXECUTION

```
//*****  
//* DSMS 2.5 *  
//*      REMPLACEMENT DES LOW-VALUES PAR DES BLANCS *  
//*****  
//$RADP.DLVB PROC ROOT='$ROOT',    RADICAL DU SYSTEME DSMS  
//      FILE='$FILE',             NUMERO DE LA BASE  
//      INDEXQ='$INDEXQ',         INDEX FICHER A GENERATION  
//      STEPLIB='$MODB',          BIBLIOTHEQUE LM BATCH  
//      OUT='$OUT',               CLASSE DE SORTIE  
//      UNITS='$UNITO',           UNITE DE SAUVEGARDE  
//      SPABB='(TRK,(10,2),RLSE)'  SPACE POUR LA SAUVEGARDE  
//*-----  
//PDSLVB EXEC PGM=PDSLVB,PARM=' '  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//SYSOUT  DD SYSOUT=&OUT  
//PACDBB  DD DSN=&INDEXQ..&ROOT.&FILE.BB(0),DISP=SHR  
//PACDB1  DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),  
//      DISP=(,CATLG,DELETE),  
//      UNIT=&UNITS,SPACE=&SPABB,  
//      DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB  
//SYSUDUMP DD SYSOUT=&OUT  
//*
```