



VisualAge Pacbase 2.5

**DSMS 2.5 BULL GCOS7, TDS  
INSTALLATION & EXPLOITATION**

DEDD7000251F

**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.ibm.com/software/ad/vapacbase/support.htm>

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

**Première Edition (Novembre 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.ibm.com/software/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

<b>1. PREAMBULE</b> .....	<b>9</b>
<b>2. LES COMPOSANTS DE DSMS</b> .....	<b>11</b>
2.1. PRESENTATION GENERALE.....	12
2.2. LES PARAMETRES DU SYSTEME.....	13
2.3. LES PROGRAMMES TP.....	19
2.4. LES PROGRAMMES BATCH.....	21
2.5. LES AUTRES BIBLIOTHEQUES.....	24
2.6. LES FICHIERS 'SYSTEME'.....	27
2.7. LES FICHIERS 'UTILISATEUR'.....	28
<b>3. ENVIRONNEMENT</b> .....	<b>31</b>
3.1. ENVIRONNEMENT DEVELOPPEMENT.....	32
3.2. ENVIRONNEMENT TP.....	33
3.3. ENVIRONNEMENT BATCH.....	36
3.4. METHODE D'ACCES.....	37
3.5. ENCOMBREMENT DES FICHIERS.....	38
<b>4. LES PROCEDURES BATCH</b> .....	<b>43</b>
4.1. PRESENTATION GENERALE.....	44
4.2. CLASSIFICATION DES PROCEDURES.....	45
4.3. ANOMALIES D'EXECUTION.....	48
<b>5. ARCHIVAGE DU JOURNAL DES MISES A JOUR (DARC)</b> .....	<b>51</b>
5.1. PRESENTATION GENERALE.....	52
5.2. ENTREES-TRAITEMENTS-RESULTATS.....	53
5.3. DESCRIPTION DES ETAPES.....	56
5.4. JCL DE LA PROCEDURE.....	58
<b>6. IMPRESSION REQUETES ET DEMANDES D'EDITION (DPRT)</b> .....	<b>59</b>
6.1. PRESENTATION GENERALE.....	60
6.2. ENTREES-TRAITEMENTS-RESULTATS.....	61
6.3. DESCRIPTION DES ETAPES.....	64
6.4. JCL DE LA PROCEDURE.....	66
6.5. LANCEMENT EN TP.....	68
<b>7. RESTAURATION (DRST)</b> .....	<b>71</b>
7.1. PRESENTATION GENERALE.....	72
7.2. ENTREES-TRAITEMENTS-RESULTATS.....	75
7.3. DESCRIPTION DES ETAPES.....	77
7.4. JCL DE LA PROCEDURE.....	80
<b>8. SAUVEGARDE (DSAV)</b> .....	<b>83</b>
8.1. PRESENTATION GENERALE.....	84
8.2. ENTREES-TRAITEMENTS-RESULTATS.....	85
8.3. DESCRIPTION DES ETAPES.....	86
8.4. JCL DE LA PROCEDURE.....	88
<b>9. REORGANISATION DES REFERENCES CROISEES (DREO)</b> .....	<b>89</b>
9.1. PRESENTATION GENERALE.....	90
9.2. ENTREES-TRAITEMENTS-RESULTATS.....	91
9.3. DESCRIPTION DES ETAPES.....	92
9.4. JCL DE LA PROCEDURE.....	94
<b>10. EXTRACTION DU JOURNAL ARCHIVE VA PAC (DEXP)</b> .....	<b>97</b>
10.1. PRESENTATION GENERALE.....	98

10.2. ENTREES-TRAITEMENTS-RESULTATS .....	99
10.3. DESCRIPTION DES ETAPES .....	101
10.4. JCL DE LA PROCEDURE.....	103
<b>11. EXTRACTION D'ENTITES (DEXT).....</b>	<b>105</b>
11.1. PRESENTATION GENERALE.....	106
11.2. ENTREES-TRAITEMENTS-RESULTATS .....	107
11.3. DESCRIPTION DES ETAPES .....	110
11.4. JCL DE LA PROCEDURE.....	112
<b>12. EXTRACTION DE TABLES POUR LISTES EXTERNES (DEXH) .....</b>	<b>115</b>
12.1. PRESENTATION GENERALE.....	116
12.2. ENTREES-TRAITEMENTS-RESULTATS .....	117
12.3. DESCRIPTION DES ETAPES .....	118
12.4. JCL DE LA PROCEDURE.....	120
<b>13. MISE A JOUR BATCH (DUPT).....</b>	<b>121</b>
13.1. PRESENTATION GENERALE.....	122
13.2. ENTREES-TRAITEMENTS-RESULTATS .....	124
13.3. DESCRIPTION DES ETAPES .....	129
13.4. JCL DE LA PROCEDURE.....	131
<b>14. INITIALISATION (DINI) .....</b>	<b>133</b>
14.1. PRESENTATION GENERALE.....	134
14.2. ENTREES-TRAITEMENTS-RESULTATS .....	135
14.3. DESCRIPTION DES ETAPES .....	137
14.4. JCL DE LA PROCEDURE.....	138
<b>15. EXTRACTION DU JOURNAL POUR MISE A JOUR (DXBJ) .....</b>	<b>139</b>
15.1. PRESENTATION GENERALE.....	140
15.2. ENTREES-TRAITEMENTS-RESULTATS .....	141
15.3. DESCRIPTION DES ETAPES .....	143
15.4. JCL DE LA PROCEDURE.....	144
<b>16. MODIFICATION DE CODES ET DE MOTS CLES (DREN) .....</b>	<b>145</b>
16.1. PRESENTATION GENERALE.....	146
16.2. ENTREES-TRAITEMENTS-RESULTATS .....	147
16.3. DESCRIPTION DES ETAPES .....	150
16.4. JCL DE LA PROCEDURE.....	152
<b>17. PRE-PROCESSEUR DAF PROGRAMMES GENERES (DPDF) .....</b>	<b>155</b>
17.1. PRESENTATION GENERALE.....	156
17.2. ENTREES-TRAITEMENTS-RESULTATS .....	157
17.3. DESCRIPTION DES ETAPES .....	159
17.4. JCL DE LA PROCEDURE.....	161
<b>18. INSTALLATION.....</b>	<b>163</b>
18.1. PRESENTATION GENERALE.....	164
18.2. BANDE D'IMPLANTATION .....	166
18.3. INSTALLATION DU JCL .....	167
18.4. DEROULEMENT DE L'IMPLANTATION .....	176
18.5. INTEGRATION DANS UN TDS EXISTANT .....	214
18.6. INSTALLATION DE L'ENVIRONNEMENT DU MODULE DAF .....	217
18.7. REIMPLANTATION .....	220
<b>19. REPRISE DSMS 8.0.1 EN DSMS 2.5 (DR80) .....</b>	<b>223</b>
19.1. OPERATIONS A EFFECTUER.....	224
19.2. ENTREE UTILISATEUR .....	225
19.3. REPRISE DE LA SAUVEGARDE .....	226
19.4. JCL DE LA PROCEDURE.....	228

<b>20. REPRISE DSMS 8.0.2 01/02 EN DSMS 2.5 (DR8Q)</b> .....	<b>231</b>
20.1. OPERATIONS A EFFECTUER.....	232
20.2. REPRISE DE LA SAUVEGARDE .....	233
20.3. JCL DE LA PROCEDURE.....	234
<b>21. REPRISE DSMS 8.0.2 COMPATIBLE AVEC VA PAC 8.0.1</b> .....	<b>235</b>
21.1. PRESENTATION.....	236
21.2. PROCEDURE 'DR8X' : ENTREE UTILISATEUR.....	237
21.3. PROCEDURE 'DR8X' .....	238
21.4. PROCEDURE 'DR8X' : JCL .....	239
<b>22. REPRISE DSMS 1.2 OU 1.5 EN DSMS 2.5</b> .....	<b>241</b>
22.1. PRESENTATION.....	242
22.2. PROCEDURE 'DR15' - DESCRIPTION DES ETAPES.....	243
22.3. PROCEDURE 'DR15' : JCL .....	244
22.4. PROCEDURE 'DR5J' - DESCRIPTION DES ETAPES .....	245
22.5. PROCEDURE 'DR5J' : JCL.....	246
<b>23. REMPLACEMENT DES LOW-VALUES PAR DES BLANCS (DLVB)</b> .....	<b>247</b>
23.1. DLVB : REMPLACEMENT DES LOW-VALUES PAR DES BLANCS.....	248
23.2. DLVB : PARAMETRES-DESCRIPTION DES ETAPES .....	249
23.3. DLVB : PROCEDURE D'EXECUTION.....	250





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.

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

## 2.2. LES PARAMETRES DU SYSTEME

### PARAMETRAGE DU MODULE DSMS

Afin de permettre à l'utilisateur d'appliquer ses normes de codification à DSMS et de répartir les fichiers sur les différents disques du site, le JCL livré lors de l'installation comporte un certain nombre de paramètres dont on trouvera une liste exhaustive ci-après.

Les paramètres sont de la forme '\$XXXXX'. Le '\$' permet de repérer le paramètre dans le nom de l'élément traité et 'XXXXX' représente son code.

La substitution de ces paramètres s'effectue globalement sur le JCL par l'exécution des procédures 'DSZZEXEC' et 'DSZZJCL' dont le mode d'utilisation est expliqué dans le sous-chapitre 'INSTALLATION DU JCL' au chapitre 'INSTALLATION'.

Dans la suite du présent manuel, on ne se référera qu'aux noms paramétrés des bibliothèques et des fichiers manipulés.

LES COMPOSANTS DE DSMS  
LES PARAMETRES DU SYSTEME

2  
2

```

COMM 'DSMS 2.5';
*****
*          PARAMETRES D'INSTALLATION          *
*
*  REMPLACEZ, SI BESOIN EST, LA VALEUR      *
*  PAR DEFAUT DE CHAQUE PARAMETRE.         *
*  LES LIGNES PARAMETRE SONT DE LA FORME:  *
*
*  $NNNNNN = VALEUR                        *
*
*  CE FICHIER EST TRAITE PAR UN             *
*  PROGRAMME EDITEUR QUI CONVERTIT        *
*  LES LIGNES PARAMETRES EN ORDRES        *
*  DE SUBSTITUTIONS.                      *
*
*  DONC,                                    *
*
*  - TOUTES LES LIGNES DONT LE PREMIER    *
*  CARACTERE NON BLANC N'EST PAS UN      *
*  DOLLAR ($) SONT TRAITEES COMME        *
*  COMMENTAIRES.                          *
*  - LE SIGNE EGAL (=) (DELIMITEUR) EST  *
*  INTERDIT DANS LA VALEUR DES          *
*  PARAMETRES.                             *
*
*****

***** UTILISATEUR D S M S PAR DEFAUT *
$USER = IBM
***** NOM DU CATALOGUE D'IMPLANTATION *
$CTNM = PV
***** NOM DU TDS      D S M S      *
$NMTD = TDS
***** BANDE D'INSTALLATION *****
$TAPE = XXXXXX
$DVTP = CT/M5
***** LANGUE E(NGLISH)/F(RANÇAIS) *****
$LANG = F
***** SUFFIXE BIBLIOTHEQUES *****
***** CU BATCH
$LIBCUB = CUBLIB
***** CU TP
$LIBCUT = CUTLIB
***** CU PERMANENTS
$LIBCUP = CUPLIB
***** JCL
$LIBJCL = JCLLIB
***** LANCEMENT JCLS INVOKE
$LIBINV = INVLIB
***** PRINT DES JCLS APPELES PAR INVOKE **
** BLANC POUR NE PAS LISTER OU &LIST POUR LISTER
$LIST = &LIST
***** LM
$LIBLM = LMLIB
***** SM (VOIR NOTE)
$LIBSM = SMLIB
***** SL TDS (VOIR NOTE)
$LIBSL = SLLIB
***** SL UTILISATEURS
$LIBSU = SULIB
***** REFERENCES FICHIERS BASE *****
***** RADICAL DES FICHIERS DSMS *
$ROOT = DS
***** IDENTIFIANT DES FICHIERS DSMS *

```

LES COMPOSANTS DE DSMS  
LES PARAMETRES DU SYSTEME

2  
2

```

$FILE      = 250
***** CISIZE DES FICHIERS BATCH          *
** VALEUR PAR DEFAUT POUR UN TYPE DE DISQUE MS/D500 **
$CISEQ    = 14336
***** REFERENCES *****
*NMXX     = PREFIXE DES FICHIERS
*DVXX     = TYPE DE DISQUE
*MDXX     = NOM DU DISQUE
*CTXX     = CATALOGAGE (Y OU N)
***** FICHIERS UTILISATEURS TDS
$NMTU     = DSTU
$DVTU     = MS/FSA
$MDTU     = DISC01
$CTTU     = Y
***** FICHIERS TDS
$DVTD     = MS/FSA
$MDTD     = DISC02
$CTTD     = Y
***** FICHIERS UTILISATEURS BATCH
$NMBU     = DSBU
$DVBU     = MS/FSA
$MDBU     = DISC03
$CTBU     = Y
***** FICHIERS BATCH SYSTEME
$NMBS     = DSBS
$DVBS     = MS/FSA
$MDBS     = DISC04
$CTBS     = Y
***** FICHIER JOURNAL DSMS
$NMDJ     = DSDJ
$DVDJ     = MS/FSA
$MDDJ     = DISC05
$CTDJ     = Y
***** FICHIER TEMPORAIRE
$DVTM     = MS/FSA
$MDTM     = DISC06
***** BIBLIOTHEQUES
$NMLI     = DSLI
$DVLI     = MS/FSA
$MDLI     = DISC07
$CTLI     = Y
***** ENVIRONNEMENT SM (VOIR NOTE) *
*** SI INSERTION DANS UN TDS EXISTANT *
*** PRENDRE DES TPRS NON ATTRIBUEES *
***** NOM DE LA SM DSMS *
$TPR0     = TPR
***** NOM DE LA SM DSMS *
$TPR1     = TPR1
***** RELEASE SYSTEME GCOS7 *
* V6 POUR RELEASE GCOS7 V6 *
* V5 POUR RELEASES ANTERIEURES *
$GCOS7    = V6
*****
* NOTE IMPORTANTE :
*****
* PAR SOUCI D'INTEGRITE AVEC LES *
* PROCEDURES LIVREES PAR LE CONSTRUCTEUR: *
* MTPREP (TP7PREP), MTGEN (TP7GEN), ETC.. *
* IBM PROPOSE UN CHOIX DE VALEURS *
* RESPECTANT LA COHERENCE DU PARAMETRAGE *
* DE CES PROCEDURES. *
* IL EST DONC FORTEMENT RECOMMANDE DE LES *
* APPLIQUER. *
*****

```

## TABLEAUX DES FICHIERS PARAMETRES

Afin de choisir la valeur des paramètres et d'estimer leur répercussion sur le nom des fichiers du système DSMS, les tableaux suivants présentent la liste de tous les fichiers regroupés par famille (premier paramètre de leur nom externe).

### LES LIBRAIRIES

-----

```
-----  
! avant paramétrage      ! avec les valeurs par défaut      !  
!-----!-----!  
! $NMLI.$LIBJCL          ! DSLI.JCLLIB                       !  
! $NMLI.$LIBSU           ! DSLI.SULIB                        !  
! $NMLI.$LIBCUB          ! DSLI.CUBLIB                       !  
! $NMLI.$LIBCUT          ! DSLI.CUTLIB                       !  
! $NMLI.$LIBCUP          ! DSLI.CUPLIB                       !  
! $NMLI.$LIBLM           ! DSLI.LMLIB                        !  
! $NMLI.SOURCE           ! DSLI.SOURCE                       !  
! $NMLI.ABOUT            ! DSLI.ABOUT                        !  
! $NMTD.$LIBSM           ! TDS.SMLIB                         !  
! $NMTD.$LIBSL           ! TDS.SLLIB                         !  
-----
```

### LES FICHIERS BATCH UTILISATEUR

-----

```
-----  
! avant paramétrage      ! avec les valeurs par défaut      !  
!-----!-----!  
! $NMBU.$ROOT$FILEBB     ! DSBU.DS200BB                      !  
! $NMBU.$ROOT$FILEBJ     ! DSBU.DS200BJ                      !  
! $NMBU.$ROOT$FILEBQ     ! DSBU.DS200BQ                      !  
-----
```



LES FICHIERS BATCH SYSTEME

-----

-----  
! avant paramétrage           ! avec les valeurs par défaut       !  
!-----!-----!  
! \$NMBS.\$ROOT\$ROOTDE0       ! DSBS.DSDSDE0                   !

LES FICHIERS TDS UTILISATEUR

-----

-----  
! avant paramétrage           ! avec les valeurs par défaut       !  
!-----!-----!  
! \$NMTU.\$ROOT\$FILEDA       ! DSTU.DS200DA                   !  
! \$NMDJ.\$ROOT\$FILEDJ       ! DSDJ.DS200DJ                   !  
! \$NMTU.\$ROOT\$FILEDX       ! DSTU.DS200DX                   !  
! \$NMTU.\$ROOT\$FILEDC       ! DSTU.DS200DC                   !  
! \$NMTU.\$ROOT\$ROOTDE       ! DSTU.DSDSDE                   !  
! \$NMTU.\$ROOT\$FILEHE       ! DSTU.DS200HE                   !  
! \$NMTU.\$ROOT\$FILEJB       ! DSTU.DS200JB                   !  
! \$NMTU.\$ROOT\$FILEPA       ! DSTU.DS200PA                   !

IMPORTANT

Dans le cas d'une installation de type CATALOGUE, il est IMPERATIF de créer au préalable les répertoires nécessaires à l'installation.

Il est nécessaire de spécifier l'option 'AUTOATTACH' pour le catalogue d'installation \$CTNM.

Si le catalogue \$CTNM n'existe pas OU s'il n'est pas auto-attachable, l'administrateur système doit IMPERATIVEMENT exécuter la procédure 'DSINMPRE' (étape n° 2 de l'installation).

## PRECISIONS SUR LE PARAMETRE \$CISEQ

Pour les fichiers utilisés dans les chaînes BATCH, le paramètre \$CISEQ permet de choisir une valeur de CISIZE afin d'optimiser l'utilisation de l'espace disque.

Ce paramètre spécifie la taille de CI en octets, il doit être multiple de 512.

Le nombre d'articles logeables dans un CI dépend du total de leurs longueurs, sachant qu'un article ne peut être réparti sur deux CI et qu'au maximum 255 articles peuvent être stockés dans un CI.

En conséquence, pour les fichiers utilisés, les valeurs de \$CISEQ proposées selon les différents types de disque sont les suivantes :

Pour un MS/FSA ----> 14336

Pour un MS/D500 ----> 14336

Pour un MS/M452 ----> 12800

Pour un MS/M500 ----> 9216

Pour un MS/D300 ----> 9216

Pour un MS/B10 ----> 12800

### \*\*\*\*\* ATTENTION \*\*\*\*\*

En version GCOS7 V3A7, la taille du buffer est égale à la taille du CI alors qu'en version GCOS7 V3B7 (mémoire paginée de 4 KBYTES), la taille du buffer est égale à la taille du CI + 32 arrondie au multiple supérieur de 4 Ko.

Pour les fichiers du TDS VA Pac, le choix d'un CISIZE de 4096 nécessiterait 2 pages mémoire (avec perte d'environ une page).

Il est donc recommandé pour cette version (GCOS7 V3B7) d'utiliser un CISIZE de 3584 pour tous les fichiers VA Pac du TDS.

## 2.3. LES PROGRAMMES TP

### LES PROGRAMMES TP

! CODE	!	CHOIX CORRESPONDANT, COMMENTAIRES	!
! PROGRAMME	!		!
! xxCHOI	!	Sous-pgm décodification choix	!
! xx0AA0	!	Première TPR	!
! PDTPDF	!	Sous-programme d'extraction DAF	!
! 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	!
! xx00Q1	!	Q . .....	!
! xx00Q2	!	Q . ..... D	!
! xx00Q3	!	LCQ	!
! xx00Q4	!	LVQ	!
! xx00Q5	!	LJQ	!
! xx00Q6	!	R . .....	!
! xx00Q7	!	R . ..... L	!
! xx00Q8	!	R ..... C	!

! CODE	!	CHOIX CORRESPONDANT, COMMENTAIRES	!
! PROGRAMME	!		!
! 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	!

Les sous-programmes TP

! CODE	!	Fonction	!
! DSCHOI	!	Décodification du choix	!
! ZAR980	!	Gestion de l'affichage	!
! ZAR985	!	Recherche du type d'écran	!
! DSCUAM	!	Contrôle TP sur fiche amélioration	!
! DSCUEV	!	Contrôle TP sur fiche événement	!
! DSCUMQ	!	Contrôle TP sur fiche maquette	!
! DSCURQ	!	Contrôle TP sur fiche requête	!
! DSCUSI	!	Contrôle TP sur fiche site	!

## 2.4. LES PROGRAMMES BATCH

### LES PROGRAMMES BATCH

! CODE	! PROC.	! Commentaire	!
! DAFD10	! DPDF	! Pré-processing DAF	!
! PACABE	!	! Abend batch	!
! 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 ctrl supplémentaires pour PDSUB1!	!
! PDSCEV	! -	! Ss-pgm ctrl supplémentaires pour PDSUE1!	!
! PDSCMQ	! -	! Ss-pgm ctrl supplémentaires pour PDSUQ6!	!
! PDSCRQ	! -	! Ss-pgm ctrl supplémentaires pour PDSUQ1!	!
! PDSCSI	! -	! Ss-pgm ctrl 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	!
! 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	! -	! -	!

! CODE	! PROC.	! Commentaire	!
! PDSR40	! DREO	! Réorganisation	!
! PDSUAA	! DUPT	! Sous-programme	!
! PDSUB1	! -	! -	!
! PDSUB2	! -	! -	!
! PDSUB3	! -	! -	!
! 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	!
!	!	!	!
! 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	!

! CODE	! PROC.	! Commentaire	!
! PDS380	! DRST	! Vérifie le journal	!
! PDS381	! -	! Vérifie le journal (version anglaise)	!
! 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	!
! PDS700	! DXBJ	! Extraction de mouvements du journal	!
! PDS900	! DUPD	! M. à j. de DSMS à partir de tables DAF!	!
!	!	!	!
! PDSR8B	! DR80	! Reprise de base DSMS 8.0 ou 8.0.1	!
! PDSR8C	!	! ATTENTION : VA PAC DOIT ETRE	!
!	!	! PREALABLEMENT BASCULE EN 8.0.2.	!
! PDSR8X	! DR8X	! Reprise de base DSMS 8.0.2	!
!	!	! compatible VA Pac 8.0.1.	!
!	!	! A UTILISER SIMULTANEMENT A LA MIGRATION!	!
!	!	! DE VA PAC 8.0.1 EN VA PAC 8.0.2	!
! PDSR8Q	! DR8Q	! Reprise de base DSMS 8.0.2 v01 ou v02	!
! PDSR8R	!	! (Reprise des Requêtes)	!
! PDSR15	! DR15	! Reprise de base DSMS 1.2 ou 1.5	!
! PDSR5J	! DR5J	! Reprise du journal archivé DSMS 1.5	!

## 2.5. LES AUTRES BIBLIOTHEQUES

### LA BIBLIOTHEQUE DU SOURCE DU TDS : \$NMTD.\$LIBSL

Sa taille est de 2 pistes de type MS/FSA. Elle contient le source TDS et le source des TPRs de début et de fin de conversation de DSMS.

```
+-----+-----+
! Membre      ! Contenu      !
+-----+-----+
! STDS        ! source TDS    !
! DS0AA0      ! source TPRs de début/fin de conversation !
+-----+-----+
```

Remarque :

Cette librairie est créée par la procédure constructeur de préparation du TDS : TP7PREP (ou MTPREP pour GCOS7 V5).

### LA BIBLIOTHEQUE DE SOURCES UTILISATEUR : \$NMLI.SOURCE

Sa taille est de 1 cylindre de type MS/FSA. Elle contient les dictionnaires DAF français et anglais (mouvements de mise à jour VisualAge Pacbase) et les sous-programmes de contrôle utilisateur pour les fiches améliorations, événements, sites, requêtes et maquettes.

```
+-----+-----+
! Membre      ! Contenu      !
+-----+-----+
! DAFDICF     ! Dictionnaire DAF pour le français      !
! DAFDICA     ! Dictionnaire DAF pour l'anglais        !
+-----+-----+
! DSCUAM      ! Contrôle TP sur fiche amélioration    !
! DSCUEV      ! Contrôle TP sur fiche événement       !
! DSCUMQ      ! Contrôle TP sur fiche maquette       !
! DSCRQ       ! Contrôle TP sur fiche requête        !
! DSCUSI      ! 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        !
+-----+-----+
```



### LA LIBRAIRIE DES MOUVEMENTS UTILISATEUR : \$NMLI.\$LIBSU

Sa taille est de 1 cylindre de type MS/FSA.

Elle contient les mouvements résultant de l'extraction.

### LA BIBLIOTHEQUE DES COMPIL-UNITS BATCH : \$NMLI.\$LIBCUB

Sa taille est d'environ 3 cylindres de type MS/FSA. Elle contient les compil-units de tous les programmes batch ainsi que les sous-programmes listés dans le sous chapitre précédent.

Sa présence sur disque n'est indispensable qu'au moment de l'implantation du module DSMS. Cette librairie est alors utilisée pour la constitution de la librairie \$NMLI.\$LIBLM.

### LA BIBLIOTHEQUE DES COMPIL-UNITS TP : \$NMLI.\$LIBCUT

Sa taille est d'environ 6 cylindres de type MS/FSA. Elle contient les compil-units de tous les programmes TP, ainsi que les sous-programmes listés ci-après.

Sa présence sur disque n'est indispensable que lors des opérations suivantes :

- . Implantation du système DSMS
- . Link des TPRS dans une SMLIB (par exemple lors de l'insertion de DSMS dans un TDS du site).

### LA BIBLIOTHEQUE DES CU TP PERMANENTS : \$NMLI.\$LIBCUP

Sa taille est de 1 cylindre de type MS/FSA. Elle ne contient que les sous-programmes DSCHOI, ZAR980 et ZAR985.

Elle ne sert que pour la génération du TDS gérant DSMS. Les compil-units des programmes ZAR980, ZAR985 et DSCHOI sont linkés lors de cette phase au TDS.

LES COMPOSANTS DE DSMS	
LES AUTRES BIBLIOTHEQUES	

2
5

### LA BIBLIOTHEQUE DE VERSIONS : \$NMLI.ABOUT

Elles contient les caractéristiques techniques de la version installée :  
nouvelles améliorations, dates des éléments, particularités d'installation.

## 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,
- . la bibliothèque des modules exécutables batch,
- . la bibliothèque des sources,
- . les bibliothèques de compil-units,
- . la bibliothèque de versions,

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

```
. Taille          : Environ 30000 enregistrements  
. Nom externe    : $NMTU.$ROOT$ROOTDE  
. Organisation   : UIND  
. Recsize       : 90  
. Clé           : 17 (position 1)  
. Csize        : 2048
```

## 2.7. LES FICHIERS 'UTILISATEUR'

### LES FICHIERS 'UTILISATEUR'

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

Les quatre premiers constituent les données directement gérées par le module. Ce sont :

.Fichier des données DSMS (DA)

-----

. Nom externe : \$NMTU.\$ROOT\$FILEDA  
. Organisation : UIND  
. Recsize : mini 80, maxi 340  
. CI size : 2048  
. Clé : 40 (position 3)

.Fichier des références croisées (DX)

-----

. Nom externe : \$NMTU.\$ROOT\$FILEDX  
. Organisation : UIND  
. Recsize : 80  
. CI size : 2048  
. Clé : 50 (position 1)

.Fichier des éléments VisualAge Pacbase (DC)

-----

. Nom externe : \$NMTU.\$ROOT\$FILEDC  
. Organisation : UIND  
. Recsize : mini 50, maxi 168  
. CI size : 4096  
. Clé : 31 (position 3)

.Fichier journal DSMS (DJ)

-----

. Nom externe : \$NMTU.\$ROOT\$FILEDJ  
. Organisation : UREL  
. Recsize : 180  
. CI size : 2048

LES COMPOSANTS DE DSMS  
LES FICHIERS 'UTILISATEUR'

2  
7

.Fichier help DSMS (HE)

-----

. Nom externe : \$NMTU.\$ROOT\$FILEHE  
. Organisation : UIND  
. Recsize : 1932  
. CI size : 4096  
. Clé : 12 (position 1)

.Fichier lanceur de job (JB)

-----

. Nom externe : \$NMTU.\$ROOT\$FILEJB  
. Organisation : UREL  
. Recsize : 80  
. CI size : 2048

.Fichier de de travail DAF TP (PA)

-----

. Nom externe : \$NMTU.\$ROOT\$FILEPA  
. Organisation : UIND  
. Recsize : mini 100, maxi 450  
. CI size : 4096  
. Clé : 37 (position 2)

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

.Fichier sauvegarde (BB)

-----

. Nom externe : \$NMBU.\$ROOT\$FILEBB(n)  
. Organisation : USEQ  
. Recsize : mini 61, maxi 350  
. CI size : 2048

.Fichier archivage du journal (BJ)

-----

. Nom externe : \$NMBU.\$ROOT\$FILEBJ(n)  
. Organisation : USEQ  
. Recsize : 180  
. CI size : 2048

.Fichier archive désactivée (BQ)

-----

. Nom externe : \$NMBU.\$ROOT\$FILEBQ  
. Organisation : USEQ  
. Recsize : 180  
. CI size : 2048

LES COMPOSANTS DE DSMS  
LES FICHIERS 'UTILISATEUR'

PAGE

30

2  
7

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
ENVIRONNEMENT

PAGE 31

3

## 3. ENVIRONNEMENT

### 3.1. ENVIRONNEMENT DEVELOPPEMENT

#### SITE D'EXPLOITATION :

Le système sur lequel fonctionne ce produit doit avoir les caractéristiques suivantes :

Systeme d'exploitation : depuis GCOS-7 V3 A7  
Niveau TDS : depuis GCOS-7 V3 A7  
Messages supportés : VIP7700, QUESTAR,  
VIP7800, IBM3270



## 3.2. ENVIRONNEMENT TP

### ENVIRONNEMENT TP

Le moniteur utilisé est TDS/GCOS 7.

Tous les fichiers 'utilisateur' sont mis à jour en TP et doivent être protégés par l'option de journalisation du TDS (journal before).

Il en est de même du fichier sauvegarde de l'écran (DS80HE) et du fichier relatif (DS80JB) nécessaire à la fonction JOB.

## GENERALITES SUR LE FONCTIONNEMENT DU SYSTEME

Les caractéristiques générales du fonctionnement du système sont les suivantes :

. Deux codes de transaction sont utilisés pour la gestion des données DSMS.

Ils pointent sur la première TPR (DS0AA0), ce programme ne gère directement aucun écran.

Les valeurs de ces codes transaction sont au choix de l'utilisateur, cependant le quatrième caractère permet de fixer le type de terminal à utiliser :

'1' : seuls les écrans VIP 7700 peuvent être utilisés

'2' : seuls les écrans IBM 3270 peuvent être utilisés

'3' : seuls les écrans VIP 7800 peuvent être utilisés

Autre : pas de restriction quant au type de terminal, qui est automatiquement géré lors de la connexion, le troisième caractère permet de fixer la langue de travail de l'utilisateur (F pour FRANCAIS ou E pour ANGLAIS).

Chaque conversation commence et s'achève avec l'exécution de la TPR 'DS0AA0'. Le source de cette TPR est livré à l'utilisateur pour lui permettre d'y inclure les traitements éventuels de début et de fin de conversation en standard sur le site ou de modifier les codes de transaction standards qui sont DSF0 et DSE0.

. En cas d'anomalie gérée par le système, celui-ci renvoie une 'MAP ABEND'. Le programme d'affichage de cet écran est appelé par un call 'ABORT'.

. Les mises à jour sont sérialisées, c'est-à-dire que le système protège les accès concurrents au réseau par mise en file d'attente des TPRs de mise à jour.

. On sort correctement du système en tapant 'FT' dans la zone option de la mire DSMS, de façon à obtenir le message 'FIN DE CONVERSATION'.

## TRAITEMENT DES MAJUSCULES ET MINUSCULES

Le système DSMS possède sa propre gestion des caractères en entrée : tous les codes introduits en minuscules sont transformés automatiquement en majuscules.

Pour bénéficier de cette gestion, il faut que l'écran soit configuré en minuscules, disposer d'une imprimante éditant des minuscules. D'autre part, avant de se connecter au TP, il faut préciser que l'on va travailler en minuscules en tapant la commande '\$\*\$LC ON' .

NOTE: En batch, la transcodification des minuscules n'est pas effectuée. Par exemple une demande d'extraction de requête saisie en minuscules n'est pas reconnue.

### 3.3. ENVIRONNEMENT BATCH

#### ENVIRONNEMENT BATCH

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

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

### 3.4. METHODE D'ACCES

#### METHODE D'ACCES

Le module DSMS gère ses fichiers à l'aide des méthodes d'accès indexée sans index secondaire et relative.

Toutes les procédures batch prévoient d'effectuer un DEALLOC PREALLOC, et éventuellement une opération de catalogage, en cas de rechargement des fichiers.

Tous les fichiers sont protégés contre les accès concurrents en écriture.

### 3.5. ENCOMBREMENT DES FICHIERS

#### ENCOMBREMENT DES FICHIERS

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

On considère que la définition globale d'un événement ou amélioration représente :

- 20 enregistrements dans le fichier des données DA (1 enregistrement fiche de définition + 19 enregistrements pour les lignes de descriptif),  
(NDA = 20),
- 10 enregistrements dans le fichier des références DX,  
(NDX = 10),
- 10 enregistrements dans le fichier des éléments PAC DC (dans le cas de maj VA Pac sous contrôle DSMS),  
(NDC = 10)
- chacun de ces enregistrements entraîne 4 mouvements de journalisation dans DJ.  
(NDJ = 4)

Exemple de calcul pour une croissance mensuelle de :

NBE: 300 événements  
NBA: 100 améliorations  
NBT: 1000 enregistrements tables (fixe)  
Sites négligeables

Soit :

NBDA le nombre d'enregistrements de DA,  
NBDX le nombre d'enregistrements de DX,  
NBDC le nombre d'enregistrements de DC,  
NBDJ le nombre d'enregistrements de DJ.

On obtient pour six mois :

$NBDA = NBT + 6 \times (NDA \times (NBE + NBA)) \rightarrow 49000$  enregistrements

ENVIRONNEMENT  
ENCOMBREMENT DES FICHIERS

PAGE

39

3  
5

$NBDX = 6 \times (NDX \times (NBE + NBA))$  --> 24000 enregistrements

$NBDC = 6 \times (NDC \times NBA)$  --> 6000 enregistrements

pour un journal de 15 jours :

$NBDJ = ((NBDA + NBDX + NBDC) / 12) \times NDJ$  --> 26340 enregistrements

## ESTIMATION DES VOLUMES

Description des fichiers :  
-----

(DA) Fichier données DSMS : 80 à 340 octets (longueur moyenne 220).

(DX) Fichier des références croisées : son volume dépend de l'utilisation des zones servant à la gestion de ces références lors de la création des événements et améliorations (phases, options, mots- clés, etc...). Contient 80 octets par enregistrement.

(DC) Fichier des éléments VA Pac : son volume dépend de l'intégration ou non de DSMS dans VA Pac. Dans le cas d'une intégration, il contient les références aux éléments VA Pac spécifiées sur les -C et -M des améliorations (50 à 168 octets par enregistrement, longueur moyenne 100).

(DJ) Fichier journal : il doit pouvoir contenir tous les mouvements de mise à jour, en batch et en TP, passés entre 2 réinitialisations de ce fichier journal, un mouvement DSMS correspondant à un enregistrement du fichier journal (de 180 octets).

### Exemple d'estimation pour six mois :

Volume du fichier des données            DA

$$\text{NBDA} \times 220 = 10.780.000 \text{ octets.}$$

Volume du fichier des références croisées DX

$$\text{NBDX} \times 80 = 1.920.000 \text{ octets.}$$

Volume du fichier des éléments VA Pac    DC

$$\text{NBDC} \times 100 = 600.000 \text{ octets.}$$

Volume du fichier journal (pour 15 jours) DJ

$$\text{NBDJ} \times 180 = 4.741.200 \text{ octets.}$$



## ENCOMBREMENT DU SYSTEME

Afin de prévoir l'espace disque nécessaire à l'installation de DSMS, les tableaux suivants montrent l'ensemble des bibliothèques et fichiers utiles à son fonctionnement, ainsi que leurs tailles (valeurs prises par défaut à l'installation).

Les exemples prévus pour l'installation représentent un environnement global d'environ 30 millions d'octets.

### LES FICHIERS SYSTEME

! Noms paramétrés	! Contenus	!Millions!
!	!	!d'octets!
!-----!	!-----!	!-----!
!\$NMLI.\$LIBLM	! Load-modules	! 3,5 !
!	!	! !
!\$NMTD.\$LIBSM	! Sharable-modules	! 4,8 !
!	!	! !
!\$NMTD.\$LIBSL	! Bibliothèque de sources	! 0,7 !
!	!	! !
!\$NMLI.\$LIBSU	! Bibli. des mvts extraits	! 0,7 !
!	!	! !
!\$NMBS.\$ROOT\$ROOTDE0	! Libellés d'erreurs DSMS	! 5,5 !
!	!	! !
!\$NMLI.\$LIBJCL	! Bibliothèque des JCLs	! 0,7 !
!	!	! !
!\$NMLI.\$LIBINV	! Bibliothèque des INVOKEs	! 0,7 !
!	!	! !
!\$NMLI.SOURCE	! Sources de contrôles	! 0,7 !
!	!	! !
!\$NMLI.ABOUT	! Bibliothèque de versions	! 0,7 !
!	!	! !
!	TOTAL	! 18 !

LES FICHIERS EVOLUTIFS

!Noms paramétrés	! Contenus	!Millions!
!	!	!d'octets!
!-----!	!-----!	!-----!
!Les bases :	!	! !
!\$NMTU.\$ROOT\$FILEDA	! Données DSMS	! 0,7 !
!\$NMTU.\$ROOT\$FILEDX	! Références croisées DSMS	! 0,7 !
!\$NMTU.\$ROOT\$FILEDC	! Eléments VA Pac	! 0,7 !
!\$NMTU.\$ROOT\$ROOTDE	! Libellés d'erreur DSMS	! 5,5 !
!	!	! !
!\$NMTU.\$ROOT\$FILEDJ	! Journal DSMS	! 0,7 !
!\$NMTU.\$ROOT\$FILEHE	! Sauvegarde écran	! 0,7 !
!\$NMTU.\$ROOT\$FILEJB	!	! 0,7 !
!	!	! !
!	!	!-----!
!	! TOTAL	! 9,7 !
!	!	!-----!
!Les sauvegardes :	!	! !
!\$NMBU.\$ROOT\$FILEBB	! Images séquentielles de la	! 0,7 !
!	! base DSMS	! !
!	!	! !
!\$NMBU.\$ROOT\$FILEBJ	! Fichiers des mouvements	! 0,7 !
!	! archivés	! !
!	!	! !
!\$NMBU.\$ROOT\$FILEBQ	! mouvements désactivés	! 0,7 !
!	!	! !
!	!	!-----!
!	! TOTAL	! 2,1 !
!	!	!-----!
!	! TOTAL	! 11,8 !
!	!	!-----!

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
LES PROCEDURES BATCH

PAGE 43

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).

## 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).

## LE JCL

Le nom des procédures se décompose en deux parties.

Les quatre premiers caractères représentent le type de la procédure :

- . DSEX procédures d'exploitation,
- . DSIN procédures d'installation,
- . DSUS procédures utilisateurs,
- . DSZZ interprétation du JCL,
- . DSIV appel des procédures par INVOKE.

Les quatre derniers caractères indiquent la procédure.

exemples :

- . DSAV sauvegarde de la base DSMS,
- . ALLO allocation des fichiers,
- . DEXT extraction d'entités.

Tous les JCLs sont paramétrés. Certains paramètres existant dans toutes les procédures n'apparaissent pas dans les notations utilisées de chaque procédure. Ce sont les noms des supports physiques.

Toutes les procédures batch sont lancées par INVOKE à partir d'un membre DSIVXXXX (librairie \$NMLI.\$LIBINV), il existe un membre de ce type par procédure.

exemple : . DSIVDRST appel de la procédure de restauration.

Ces membres contiennent un appel du jcl à lancer (INVOKE) et pour les procédures avec entrées utilisateur, le contenu de celles-ci dans des cartes data de 80 caractères délimitées par \$INPUT et \$ENDINPUT.

### 4.3. ANOMALIES D'EXECUTION

#### ANOMALIES D'EXECUTION

Les erreurs entrée-sortie sur les fichiers du système ou de la base provoquent la sortie d'un état d'anomalie (PAC7EI) édité plusieurs fois, le positionnement du switch-20 et le débranchement vers la fin de la procédure.

Dans la plupart des cas, l'examen du switch-20 et le type d'opération effectuée permettent de trouver la cause de la fin anormale (ressources non disponibles, fichier trop petit, etc...).

Si l'état PAC7EI est absent et que le type de l'ABEND met en cause directement les programmes du système DSMS, il est nécessaire de contacter l'équipe technique AD-TOOLS d'IBM et de conserver tous les listings.

L'édition de l'état PAC7EI est assurée par le sous-programme PACABE. L'utilisateur pourra mettre en place à la fin de chacune des procédures un traitement d'erreur particulier conditionné par la valeur du switch-20.

Du fait de leur caractère systématique, le positionnement du switch-20 et le débranchement ne seront pas indiqués dans les descriptions des procédures ni dans les dessins de chaînes.

#### MESSAGES DU COMPTE RENDU D'EXECUTION

Certains programmes de DSMS produisent le message suivant dans le compte rendu d'exécution :

```
WARNING EX02. DECIMAL DATA OVERFLOW IN TASK xxxx AT  
ADDRESS  
x.xx.xxxx
```

Rappelons que ce message est édité lorsque des zones numériques sont tronquées à gauche de chiffres significatifs. Cet avertissement ne correspond donc pas à une anomalie d'exécution et dans le cas de DSMS, certaines méthodes de programmation impliquent l'édition de ce message.

Il convient donc de ne pas attacher d'importance à l'édition de ce type de



message lors de l'exécution des programmes de DSMS et de contacter IBM si le compteur de warning est dépassé.

### GESTION DES ERREURS DANS LES PROCEDURES LIVREES

A la fin de chaque procédure, la détection d'une erreur provoque l'arrêt de la procédure avec un code retour différent de zéro. Ce code est récupérable dans la variable \$? tout de suite après la commande de lancement de la procédure.

Ceci permet d'empêcher l'exécution d'une suite si plusieurs procédures sont enchaînées.

LES PROCEDURES BATCH  
ANOMALIES D'EXECUTION

PAGE

50

4  
3

VISUALAGE PACBASE - MANUEL D'EXPLOITATION	PAGE	51
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

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Archivage du journal : PDS300
- . Définition du fichier journal : DEALLOC/PREALLOC
- . Réinitialisation du journal : PDS320
- . Rotation des générations de BJ : SHIFT

PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDARC  
(reclsize 80, Csize 2048)

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  
fichier temporaire TMBDARC
  - Archive antérieure  
PACDJB : Nom externe \$NMBU.\$ROOT\$FILEBJ
  - Fichier journal à réinitialiser  
PACDDJ : Nom externe \$NMTU.\$ROOT\$FILEDJ
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE



- . Fichier en entrée-sortie :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
- . Fichiers en sortie :
  - Archive mise à jour  
PACDBJ : Nom externe \$NMBU.\$ROOT\$FILEBJ/G+1
  - Archive désactivée  
PACDBQ : Nom externe \$NMBU.\$ROOT\$FILEBQ
- . Etat en sortie :
  - Compte-rendu d'archivage
- . 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 : DEALLOC/PREALLOC  
-----

Cette étape effectue un DEALLOC/PREALLOC du journal (DJ).

REINITIALISATION DU JOURNAL : PDS320

Cette étape effectue 2 types de traitements :

- . Création d'un enregistrement dans le fichier journal
- . Dépositionnement du TOP du fichier des données
- . Fichiers en entrée :
  - Mouvement utilisateur  
fichier temporaire TMBDARC
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Fichier en entrée-sortie :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
- . Fichier en sortie :
  - Fichier journal à réinitialiser  
PACDDJ : Nom externe \$NMTU.\$ROOT\$FILEDJ
- . Etat en sortie :
  - Compte-rendu de réinitialisation

ROTATION DES GENERATIONS DU FICHIER ARCHIVE : SHIFT  
-----

Cette étape effectue une rotation des générations du fichier archive (BJ).

## 5.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '* D S M S : ARCHIVAGE DU JOURNAL DSMS *';
COMM '* *';
COMM '* DSEXDARC *';
COMM '* ===== *';
COMM '* *';
COMM '*****';
MVL SIZEWK=2,
    CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    RFTU=&CTTU$CTTU,
    CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
    RFBS=&CTBS$CTBS,
    CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
    RFBU=&CTBU$CTBU,
    CTDJN=' FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
    RFDJ=&CTDJ$CTDJ,
    RFTM=' DVC=$DVTM,MD=$MDTM';
CR IF=*DARC,
    OF=(TMBDARC,&RFTM,TEMPRY,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP PDS300,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
    SZ 100;
    ASG PACDMB,TMBDARC,TEMPRY,&RFTM,END=PASS;
    ASG PACDDA,$NMTU.$ROOT$FILEDA,&RFTU;
    ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU;
    ASG PACDDJ,$NMDJ.$ROOT$FILEDJ,&RFDJ;
    ASG PACDJB,$NMBU.$ROOT$FILEBJ,&RFBU;
    ASG PACDBJ,$NMBU.$ROOT$FILEBJ/G+1,&RFBU;
    ASG PACDBQ,$NMBU.$ROOT$FILEBQ,&RFBU;
    ASG PACDRU,SYS.OUT;
    ASG PAC7EI,SYS.OUT;
    SWK WKDISK=(SZ=&SIZEWK,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP ERR,SW30,EQ,1;
COMM '*** ALLOCATION : DJ ***';
IV DSINALDJ ($NMLI.$LIBJCL,&RFLI);
STEP PDS320,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
    ASG PACDMB,TMBDARC,TEMPRY,&RFTM;
    ASG PACDDJ,$NMDJ.$ROOT$FILEDJ,&RFDJ;
    ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU;
    ASG PACDDA,$NMTU.$ROOT$FILEDA,&RFTU;
    ASG PACDRU,SYS.OUT;
    ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
SHIFT $NMBU.$ROOT$FILEBJ;
JUMP END;
ERR:
SEND ' DSEXDARC - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:

```

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
IMPRESSION REQUETES ET DEMANDES D'EDITION (DPRT)

PAGE 59

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 ! 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'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 ! uuuuuuuu ! 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 !uuuuuu! 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 : !
!      !      !      ! (99 requêtes maximum) !
!  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 :

#### NOTATIONS UTILISEES

```
-----  
! NOTATION ! SIGNIFICATION <EXEMPLE> !  
!-----!  
! &SIZEW1 ! réservé pour les fichier temporaires 01 !  
! &SIZEW2 ! réservé pour les fichier temporaires 01 !  
! &SIZEW3 ! réservé pour les fichier temporaires 01 !  
! &SIZEW4 ! réservé pour les fichier temporaires 01 !  
! &SIZEWK ! réservé pour les tris internes 01 !  
-----
```

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Edition : PDSB

#### PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDPRT  
(reclsize 80, Csize 2048)

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

#### EDITIONS : PDSB

- . Fichiers permanents en entrée :
  - Fichier des données  
DS80IA : Nom externe : \$NMTU.\$ROOT\$FILEDA
  - Fichier des éléments VisualAge Pacbase  
PB80DC : Nom externe : \$NMTU.\$ROOT\$FILEDC
  - Fichier des libellés d'erreur  
DS80IE : Nom externe : \$NMTU.\$ROOT\$ROOTDE



- . Fichier en entrée :
  - Requetes utilisateur  
PACDMB : TMBDPRT
  
- . Fichiers de travail :
  - Demandes d'edition  
PACDKD
  - Requetes  
PACDKQ
  - Temporaires  
PACDQR PACDQJ  

PACDW1 PACDW2  
PACDW3 PACDW4
  
- . 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

```

COMM '*****';
COMM '*   D S M S       : EDITIONS          *';
COMM '*                                           *';
COMM '*                               DSUSDPRT      *';
COMM '*                               =====      *';
COMM '*                                           *';
COMM '*****';
MVL  SIZEWK=5,
      SIZEW1=01,SIZEW2=01,SIZEW3=01,SIZEW4=01,
      CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      RFBU=&CTBU$CTBU,
      RFTM=' DVC=$DVTM,MD=$MDTM',
      PGE=' E',PGLNG=&PG$LANG;
CR   IF=*DPRT,
      OF=(TMBDPRT,&RFTM,TEMPRY,END=PASS),
      OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PDSB ***';
STEP PDSB&PGLNG,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
      SZ 500;
      ASG PACDMB,TMBDPRT,TEMPRY,&RFTM;
      ASG DS80IA,$NMTU.$ROOT$FILEDA,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF DS80IA,READLOCK=STAT;
      ASG PB80DC,$NMTU.$ROOT$FILEDC,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF PB80DC,READLOCK=STAT;
      ASG DS80IE,$NMTU.$ROOT$ROOTDE,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF DS80IE,READLOCK=STAT;
      ASG PACDW1,TPACDW1,TEMPRY,&RFTM,END=PASS;
      ALC PACDW1,SZ=&SIZEW1,INCRSZ=1,UNIT=CYL;
      DEF PACDW1,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDW2,TPACDW2,TEMPRY,&RFTM,END=PASS;
      ALC PACDW2,SZ=&SIZEW2,INCRSZ=1,UNIT=CYL;
      DEF PACDW2,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDW3,TPACDW3,TEMPRY,&RFTM,END=PASS;
      ALC PACDW3,SZ=&SIZEW3,INCRSZ=1,UNIT=CYL;
      DEF PACDW3,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDW4,TPACDW4,TEMPRY,&RFTM,END=PASS;
      ALC PACDW4,SZ=&SIZEW4,INCRSZ=1,UNIT=CYL;
      DEF PACDW4,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDKD,TPACDKD,TEMPRY,&RFTM,END=PASS;
      ASG PACDKQ,TPACDKQ,TEMPRY,&RFTM,END=PASS;
      ASG PACDQR,TPACDQR,TEMPRY,&RFTM,END=PASS;
      ASG PACDQJ,TPACDQJ,TEMPRY,&RFTM,END=PASS;
      ASG PACDIA,SYS.OUT;
      ASG PACDIB,SYS.OUT;
      ASG PACDID,SYS.OUT;
      ASG PACDIQ,SYS.OUT;
      ASG PACDJQ,SYS.OUT;
      ASG PACDQI,SYS.OUT;
      ASG PACDRQ,SYS.OUT;
      ASG PAC7EI,SYS.OUT;
      SWK WKDISK=(SZ=&SIZEWK,&RFTM);
  
```

IMPRESSION REQUETES ET DEMANDES D'EDITION (DPRT)  
JCL DE LA PROCEDURE

PAGE

67

6  
4

```
ESTP;  
JUMP ERR,SW20,EQ,1;  
JUMP END;  
ERR:  
SEND ' DSUSDPRT - ABNORMAL END OF RUN (I/O ERROR) ' ;  
LET SEV 3;  
END:
```

## 6.5. LANCEMENT EN TP

### UTILISATION DE LA FONCTION JOB

La soumission du flot de JCL constitué avec la commande 'JOB' est indirecte. Elle passe par la mise à jour d'un fichier tampon UFAS relatif (DS80JB) et par l'exécution d'un 'LANCEUR' paramétré.

Le premier enregistrement du fichier tampon est constitué de l'identification du lanceur (sous-fichier d'une bibliothèque) et du numéro du dernier enregistrement écrit dans le fichier tampon.

A la demande 'JOB', DSMS lit le premier enregistrement de ce fichier, afin de connaître l'adresse du dernier enregistrement écrit ; le flot de JCL est constitué et est écrit à la suite dans le fichier tampon. Le premier enregistrement du fichier tampon est mis à jour avec l'adresse du dernier enregistrement écrit.

DSMS fait ensuite appel à la routine système "SUBJOB" et envoie le LANCEUR pour exécution, avec les paramètres :

START = le numéro du premier enregistrement du train dans le tampon.

HALT = le nombre d'enregistrements qui constituent le train.

USER = le code DSMS de l'utilisateur qui fait la demande.

SYSUSER = Code IOF de l'utilisateur.

A ce niveau, DSMS rend la main à l'utilisateur.

Le LANCEUR effectue deux étapes.

- 1) A l'aide de l'utilitaire CREATE, il reconstitue le flot grâce aux paramètres START et HALT et crée un fichier temporaire qui ne contient que le JCL.
- 2) Par la commande RUN, il fait exécuter le JOB en lui transmettant le paramètre USER.

EXEMPLE DE JCL SAISI SOUS DSMS PAR UN UTILISATEUR

ECRAN DSMS OPTION : C1 CHOIX : LJQ

```
T NLG : V LIBELLE
A 050 : V $JOB DSMSDPRT;
A 060 : V VL DV='DVC=$DVLI',MD='$MDLI';
A 100 : V SEND '==> DEBUT DE REQUETE DSMS <==' &USER;
Z 120 : V IV DSUSDPR ( $NMLI.$LIBJCL,&MD,&DV);
A 900 : V $INPUT DPRT TYPE=DATA;
Z 100 : V $ENDINPUT DPRT;
Z 200 : V SEND '==> FIN DE REQUETE DSMS <==' &USER;
Z 300 : V $ENDJOB;
```

IMPRESSION REQUETES ET DEMANDES D'EDITION (DPRT)  
LANCEMENT EN TP

PAGE

70

6  
5

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
RESTAURATION

(DRST)

PAGE 71

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.

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.



## SOUS-PROGRAMMES DE CONTROLE DES DEFINITIONS

Ces sous-programmes (livrés sous forme de sources COBOL) permettent d'ajouter des contrôles spécifiques et de substituer des valeurs sur les 5 écrans de type " fiche " (amélioration, événement, maquette, requête et site).

Processus :

Au premier affichage de l'écran, aucun accès n'est effectué au sous-programme.

A la réception, les contrôles habituels sont tout d'abord exécutés par l'écran fiche puis le sous-programme est appelé. Celui-ci cherchera des erreurs dites " fatales " (F40) et enverra éventuellement un message accompagné d'un blocage de la mise à jour au programme appelant qui se contentera d'afficher.

Si aucune erreur de ce type n'est détectée (ou après correction par l'utilisateur suivi des contrôles usuels et d'un nouvel appel au sous-programme), un second examen des zones saisies pourra provoquer un avertissement (F45). Il suffira alors à l'utilisateur de taper ENTER pour prendre en compte la valeur précédemment saisie.

Ensuite, par enchaînement ou nouvel appel, le sous-programme pourra affecter une nouvelle valeur à certaines zones saisissables (F50).

Lors du retour au programme appelant, toutes les valeurs (saisies par l'utilisateur ou attribuées par le sous-programme) seront à nouveau contrôlées. La base pourra alors être mise à jour par cet écran.

Ces sources ne comportent au départ que 3 exemples :

- 1 avertissement,
- 1 erreur sévère,
- 1 valeur fixe

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

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

Ils sont au nombre de 10 : 5 sont prévus pour les traitements effectués en TP et 5 pour le batch (DUPT)

REMARQUE :

-Les positionnements d'erreurs se font par l'intermédiaire des "PR" (comme dans VAPAC); ces zones doivent être alimentées par 'W' ou 'E'.

## 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

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Contrôle du contenu du journal : PDS380
- . Définition des fichiers : DEALLOC/PREALLOC
- . Restauration des fichiers : PDS400
- . Réapplication de l'archive : PDS450

#### PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDRST  
(reysize 80, Csize 2048)

#### 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  
PACDDJ : Nom externe &INDUV..&ROOTD.\$ROOT2.0DJ
  - Fichier des libellés d'erreur  
PACCDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Etat en sortie
  - Etat du fichier AJ  
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 : DEALLOC/PREALLOC

Cette étape n'est exécutée que si le fichier journal a été archivé. Elle effectue un DEALLOC/PREALLOC des fichiers.

- . Fichiers définis :
  - Fichier des données  
Nom externe : \$NMTU.\$ROOT\$FILEDA
  - Fichier des éléments VA PAC  
Nom externe : \$NMTU.\$ROOT\$FILEDC
  - Fichier journal  
Nom externe : \$NMTU.\$ROOT\$FILEDJ
  - Fichier des références croisées  
Nom externe : \$NMTU.\$ROOT\$FILEDX

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 : Nom externe \$NMBU.\$ROOT\$FILEBB
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Fichiers permanents en sortie :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des éléments VA PAC  
PACDDC : Nom externe \$NMTU.\$ROOT\$FILEDC
  - Fichier journal  
PACDDJ : Nom externe \$NMTU.\$ROOT\$FILEDJ
  - Fichier des références croisées  
PACDDX : Nom externe \$NMTU.\$ROOT\$FILEDX
- . Fichier mouvement en entrée :
  - Mouvements utilisateurs  
PACDMB : Fichier temporaire TMBDRST
- . Fichier en sortie :
  - Fichier de travail (2 enreg.)  
PACDMS : Fichier temporaire TPACDMS (recsize 80)
- . 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  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des éléments VA PAC  
PACDDC : Nom externe \$NMTU.\$ROOT\$FILEDC
  - Fichier des références croisées  
PACDDX : Nom externe \$NMTU.\$ROOT\$FILEDX
- . Fichiers en entrée :
  - Fichier de travail (2 enreg.)

RESTAURATION  
DESCRIPTION DES ETAPES

(DRST)

PAGE

79

7  
3

PACDMS : Fichier temporaire TPACDMS (recsize 80)

-Fichier des libellés d'erreur

PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE

. Fichier archive en entrée :

-Archivage journal à réappliquer

PACDBJ : om externe \$NMBU.\$ROOT\$FILEBJ

. Etat en sortie :

-Compte-rendu de mise à jour

PACDRU

## 7.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '*          DSMS : CHARGEMENT-RESTAURATION BASE DSMS *';
COMM '*                                           *';
COMM '*          DSEXDRST                                           *';
COMM '*          =====                                           *';
COMM '*                                           *';
COMM '*****';
MVL  CTTUN= ' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU ' ,
      RFTU=&CTTU$CTTU ,
      CTBSN= ' FILESTAT=UNCAT ,DVC=$DVBS ,MD=$MDBS ' ,
      RFBS=&CTBS$CTBS ,
      CTLIN= ' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      CTBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
      RFBU=&CTBU$CTBU ,
      CTDJN= ' FILESTAT=UNCAT ,DVC=$DVDJ ,MD=$MDDJ ' ,
      RFDJ=&CTDJ$CTDJ ,
      RFTM= ' DVC=$DVTM ,MD=$MDTM ' ,
      PGE= ' 1 ' , PGF= ' 0 ' , PGLNG=&PG$LANG;
CR   IF=*DRST ,
      OF=( TMBDRST , TEMPRY , &RFTM , END=PASS ) ,
      OUTDEF=( CISZ=2048 , RECSZ=80 , RECFORM=FB );
COMM '*** CTL SUR DJ ***';
FILLIST INFILE=( $NMDJ . $ROOT$FILEDJ , &RFDJ );
JUMP CRE SEV GE 3;
COMM '*** PDS380 ***';
STEP PDS38&PGLNG , FILE=( $NMLI . $LIBLM , &RFLI ) , DUMP=DATA;
      SZ 30;
      ASG PACDDJ , $NMDJ . $ROOT$FILEDJ , &RFDJ;
      ASG PACDDE , $NMTU . $ROOT$ROOTDE , &RFTU ,
          ACCESS=READ , SHARE=MONITOR;
      DEF PACDDE , READLOCK=STAT;
      ASG PACDRU , SYS . OUT;
      ASG PAC7EI , SYS . OUT;
ESTP;
JUMP ERR , SW20 , EQ , 1;
JUMP END , SW30 , EQ , 1;
CRE:
COMM '*** ALLOCATION : DA , DC , DJ , DX ***';
IV  DSINALDA ( $NMLI . $LIBJCL , &RFLI );
IV  DSINALDC ( $NMLI . $LIBJCL , &RFLI );
IV  DSINALDJ ( $NMLI . $LIBJCL , &RFLI );
IV  DSINALDX ( $NMLI . $LIBJCL , &RFLI );
COMM '*** PDS400 ***';
STEP PDS400 , FILE=( $NMLI . $LIBLM , &RFLI ) , DUMP=DATA;
      SZ 60;
      ASG PACDMB , TMBDRST , TEMPRY , &RFTM;
      ASG PACDDJ , $NMDJ . $ROOT$FILEDJ , &RFDJ;
      ASG PACDBB , $NMBU . $ROOT$FILEBB , &RFBU;
      ASG PACDDC , $NMTU . $ROOT$FILEDC , &RFTU;
      ASG PACDDA , $NMTU . $ROOT$FILEDA , &RFTU;
      ASG PACDDX , $NMTU . $ROOT$FILEDX , &RFTU;
      ASG PACDDE , $NMTU . $ROOT$ROOTDE , &RFTU ,
          ACCESS=READ , SHARE=MONITOR;
      DEF PACDDE , READLOCK=STAT;
      ASG PACDMS , TPACDMS , TEMPRY , &RFTM , END=PASS;
      ASG PACDRU , SYS . OUT;
      ASG PAC7EI , SYS . OUT;
ESTP;
JUMP ERR , SW20 , EQ , 1;

```



```
JUMP END,SW30,EQ,1;
COMM '*** PDS450 ***';
STEP PDS450,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
SZ 60;
ASG PACDMS,TPACDMS,TEMPRY,&RFTM;
ASG PACDDA,$NMTU.$ROOT$FILEDA,&RFTU;
ASG PACDDC,$NMTU.$ROOT$FILEDC,&RFTU;
ASG PACDDX,$NMTU.$ROOT$FILEDX,&RFTU;
ASG PACDBJ,$NMBU.$ROOT$FILEBJ,&RFBU;
ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU,
ACCESS=READ,SHARE=MONITOR;
DEF PACDDE,READLOCK=STAT;
ASG PACDRU,SYS.OUT;
ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' DSEXDRST - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:
```

RESTAURATION  
JCL DE LA PROCEDURE

(DRST)

PAGE

82

7  
4

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
SAUVEGARDE

(DSAV)

PAGE 83

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

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Vérification intégrité des données : PDSBAS
- . Sauvegarde de la base DSMS : PDS500
- . Rotation des générations de BJ : SHIFT

PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDSAV  
(reysize 80, Cysize 2048)

VERIFICATION INTEGRITE DE LA BASE : PDSBAS

- . Fichiers permanents en entrée :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Etat en sortie :
  - Compte rendu de validité  
PACDRS
- . Code retour :
  - Switch 30 0: base valide
  - 1: base invalide. Arrêt de la procédure.

SAUVEGARDE DE LA BASE : PDS500

- . Fichier en entrée-sortie :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA

- . Fichiers permanents en entrée :
  - Fichier des éléments VisualAge Pacbase  
PACDDC : Nom externe \$NMTU.\$ROOT\$FILEDC
  - Fichier des références croisées  
PACDDX : Nom externe \$NMTU.\$ROOT\$FILEDX
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
  
- . Fichier mouvement en entrée :
  - Mouvements utilisateurs  
PACDDB : fichier temporaire TMBDSAV
  
- . Fichier en sortie :
  - Image séquentielle des fichiers  
Nom externe \$NMBU.\$ROOT\$FILEBB/G+1
  
- . Etat en sortie :
  - Compte rendu de sauvegarde

ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier archive (BB).

## 8.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '*          DSMS      : SAUVEGARDE DE LA BASE DSMS      *';
COMM '*                                                    *';
COMM '*                  DSEXDSAV                            *';
COMM '*                  =====                            *';
COMM '*                                                    *';
COMM '*****';
MVL  CTTUN= ' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU ' ,
      RFTU=&CTTU$CTTU ,
      CTBSN= ' FILESTAT=UNCAT ,DVC=$DVBS ,MD=$MDBS ' ,
      RFBS=&CTBS$CTBS ,
      CTLIN= ' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      CTBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
      RFBU=&CTBU$CTBU ,
      RFTM= ' DVC=$DVTM ,MD=$MDTM ' ;
CR   IF=*DSAV ,
      OF= ( TMBDSAV , TEMPRY , &RFTM , END=PASS ) ,
      OUTDEF= ( CISZ=2048 , RECSZ=80 , RECFORM=FB ) ;
COMM '*** PDSBAS ***';
STEP PDSBAS , FILE= ( $NMLI . $LIBLM , &RFLI ) , DUMP=DATA ;
      SZ 30 ;
      ASG PACDDA , $NMTU . $ROOT$FILEDA , &RFTU ;
      ASG PACDDE , $NMTU . $ROOT$ROOTDE , &RFTU ;
      ASG PACDRS , SYS . OUT ;
      ASG PAC7EI , SYS . OUT ;
ESTP ;
JUMP ERR , SW20 , EQ , 1 ;
JUMP END , SW30 , EQ , 1 ;
COMM '*** PDS500 ***';
STEP PDS500 , FILE= ( $NMLI . $LIBLM , &RFLI ) , DUMP=DATA ;
      SZ 60 ;
      ASG PACDDA , $NMTU . $ROOT$FILEDA , &RFTU ;
      ASG PACDDC , $NMTU . $ROOT$FILEDC , &RFTU ;
      ASG PACDDX , $NMTU . $ROOT$FILEDX , &RFTU ;
      ASG PACDDE , $NMTU . $ROOT$ROOTDE , &RFTU ;
      ASG PACDBB , $NMBU . $ROOT$FILEBB / G + 1 , &RFBU ;
      ASG PACDMB , TMBDSAV , TEMPRY , &RFTM ;
      ASG PACDRU , SYS . OUT ;
      ASG PAC7EI , SYS . OUT ;
ESTP ;
JUMP ERR , SW20 , EQ , 1 ;
SHIFT $NMBU . $ROOT$FILEBB ;
JUMP END ;
ERR :
SEND ' DSEXDSAV - ABNORMAL END OF RUN ( I / O ERROR ) ' ;
LET SEV 3 ;
END :

```



VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
REORGANISATION DES REFERENCES CROISEES (DREO)

PAGE 89

9

## **9. REORGANISATION DES REFERENCES CROISEES (DREO)**

## 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

#### NOTATIONS UTILISEES

```

-----
! NOTATION ! SIGNIFICATION                                <EXEMPLE>!
!-----!
! &SIZEW1 ! réservé pour les fichier temporaires        01 !
! &SIZEW2 ! réservé pour les fichier temporaires        01 !
! &SIZEW3 ! réservé pour les fichier temporaires        01 !
! &SIZEW4 ! réservé pour les fichier temporaires        01 !
! &SIZEW5 ! réservé pour les fichier temporaires        01 !
! &SIZEW6 ! réservé pour les fichier temporaires        01 !
! &SIZEWK ! réservé pour les tris internes              01 !
-----
    
```

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Constitution index non mots-clés : PDSR10
- . Constitution index mots-clés : PDSR20
- . Fusion des index : PDSR30
- . Fusion générale pour sauvegarde : PDSR40
- . Rotation des générations de BB : SHIFT

PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDREO  
(recline 80, Csize 2048)

CONSTITUTION INDEX (non mots-clés) : PDSR10

- . Fichier en entrée :
  - Mouvements
    - CARTE : fichier temporaire TMBDREO
- . Fichiers permanents en entrée :
  - Sauvegarde de la base DSMS
    - PACDDB : Nom externe \$NMBU.\$ROOT\$FILEBB
  - Fichier des libellés d'erreur
    - PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Fichiers de travail

- Données et éléments VA Pac  
PACDW1 :
- Mots-clés et Références mots-clés  
PACDW2 :
- Références croisées non mots-clés  
PACDW3 :
- Fichiers de tri
  
- . 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  
TPACDW2
  - Mots-clés  
TPACDW4
  - Références mots-clés  
TPACDW5
  - Fichiers de tri

FUSION DES INDEX : PDSR30

- . Fichiers de travail
  - Références croisées non mots-clés  
TPACDW3
  - Références mots-clés  
TPACDW5
  - Références mots-clés  
TPACDW6
  - Fichiers de tri

FUSION GENERALE POUR SAUVEGARDE : PDSR40

- . Fichiers de travail
  - Données et éléments VA Pac  
TPACDW1
  - Mots-clés  
TPACDW4
  - Références mots-clés  
TPACDW6
  - Fichiers de tri
  
- . Fichier permanent en entrée :
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
  
- . Fichier permanent en sortie :
  - Sauvegarde base DSMS réorganisée  
PACDBB : Nom externe \$NMBU.\$ROOT\$FILEBB/G+1
  
- . Etat en sortie :
  - Compte-rendu de réorganisation  
PACDRR

ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier archive (BB).

## 9.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '*          DSMS      : REORGANISATION DE LA BASE DSMS      *';
COMM '*                                                    *';
COMM '*                  DSEXDRIO                               *';
COMM '*                  =====                               *';
COMM '*                                                    *';
COMM '******';
MVL  SIZEW1=01,SIZEW2=01,SIZEW3=01,SIZEW4=02,SIZEW5=01,SIZEW6=01,
      SIZEWK=01,
      CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      RFBU=&CTBU$CTBU,
      RFTM=' DVC=$DVTM,MD=$MDTM';
CR   IF=*DREO,
      OF=( TMBDREO,&RFTM,TEMPRY,END=PASS),
      OUTDEF=( CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PDSR10 ***';
STEP PDSR10,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
      SZ 60;
      ASG CARTE,TMBDREO,TEMPRY,&RFTM;
      ASG PACDBB,$NMBU.$ROOT$FILEBB,&RFBU;
      ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF PACDDE,READLOCK=STAT;
      ASG PACDW1,TPACDW1,TEMPRY,&RFTM,END=PASS;
      ALC PACDW1,SZ=&SIZEW1,UNIT=CYL;
      DEF PACDW1,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDW2,TPACDW2,TEMPRY,&RFTM,END=PASS;
      ALC PACDW2,SZ=&SIZEW2,UNIT=CYL;
      DEF PACDW2,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDW3,TPACDW3,TEMPRY,&RFTM,END=PASS;
      ALC PACDW3,SZ=&SIZEW3,UNIT=CYL;
      DEF PACDW3,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDRK,SYS.OUT;
      ASG PACDRH,SYS.OUT;
      ASG PAC7EI,SYS.OUT;
      SWK WKDISK=( SZ=&SIZEWK,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PDSR20 ***';
STEP PDSR20,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
      SZ 60;
      ASG PACDW2,TPACDW2,TEMPRY,&RFTM,END=PASS;
      DEF PACDW2,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDW4,TPACDW4,TEMPRY,&RFTM,END=PASS;
      ALC PACDW4,SZ=&SIZEW4,UNIT=CYL;
      DEF PACDW4,CISIZE=$CISEQ,NBBUF=1;
      ASG PACDW5,TPACDW5,TEMPRY,&RFTM,END=PASS;
      ALC PACDW5,SZ=&SIZEW5,UNIT=CYL;
      DEF PACDW5,CISIZE=$CISEQ,NBBUF=1;
      ASG PAC7EI,SYS.OUT;
      SWK WKDISK=( SZ=&SIZEWK,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PDSR30 ***';

```

```
STEP PDSR30, FILE=( $NMLI.$LIBLM, &RFLI ), DUMP=DATA;
  SZ 60;
  ASG PACDW3, TPACDW3, TEMPRY, &RFTM, END=PASS;
  DEF PACDW3, CISIZE=$CISEQ, NBBUF=1;
  ASG PACDW5, TPACDW5, TEMPRY, &RFTM, END=PASS;
  DEF PACDW5, CISIZE=$CISEQ, NBBUF=1;
  ASG PACDW6, TPACDW6, TEMPRY, &RFTM, END=PASS;
  ALC PACDW6, SZ=&SIZEW6, UNIT=CYL;
  DEF PACDW6, CISIZE=$CISEQ, NBBUF=1;
  ASG PAC7EI, SYS.OUT;
  SWK WKDISK=( SZ=&SIZEWK, &RFTM );
ESTP;
JUMP ERR, SW20, EQ, 1;
COMM '*** PDSR40 ***';
STEP PDSR40, FILE=( $NMLI.$LIBLM, &RFLI ), DUMP=DATA;
  SZ 60;
  ASG PACDDE, $NMTU.$ROOT$ROOTDE, &RFTU,
    ACCESS=READ, SHARE=MONITOR;
  DEF PACDDE, READLOCK=STAT;
  ASG PACDBB, $NMBU.$ROOT$FILEBB/G+1, &RFBU;
  ASG PACDW1, TPACDW1, TEMPRY, &RFTM;
  DEF PACDW1, CISIZE=$CISEQ, NBBUF=1;
  ASG PACDW4, TPACDW4, TEMPRY, &RFTM;
  DEF PACDW4, CISIZE=$CISEQ, NBBUF=1;
  ASG PACDW6, TPACDW6, TEMPRY, &RFTM;
  DEF PACDW6, CISIZE=$CISEQ, NBBUF=1;
  ASG PACDRR, SYS.OUT;
  ASG PAC7EI, SYS.OUT;
  SWK WKDISK=( SZ=&SIZEWK, &RFTM );
ESTP;
JUMP ERR, SW20, EQ, 1;
SHIFT $NMBU.$ROOT$FILEBB;
JUMP END;
ERR:
SEND ' DSEXDRIO - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:
```

REORGANISATION DES REFERENCES CROISEES  
JCL DE LA PROCEDURE

(DREO)

PAGE

96

9  
4



VISUALAGE PACBASE - MANUEL D'EXPLOITATION	PAGE	97
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 ! SSAAMJJ  ! Date de début de la sélection      !  
!    !   ! 'JOUR'   ! Date de début = date du jour       !  
! 20 !   8 ! SSAAMJJ  ! 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

#### NOTATIONS UTILISEES

```
-----  
! NOTATION ! SIGNIFICATION <EXEMPLE>!  
!-----!  
! &PAC7PJ ! nom du fichier journal VA Pac archivé !  
! &SIZEWK ! réservé au tri interne 01 !  
! &USER ! nom de l'utilisateur DSMS IBM !  
-----
```

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Edition : PDS600
- . Edition des mouvements de mise à jour : PDS610

#### PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDEXP  
(recline 80, Csize 2048)

#### EXTRACTION ET FORMATAGE DE MOUVEMENTS : PDS600

- . Fichiers permanents en entrée :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
  - Journal archivé VA Pac  
PAC7PJ Nom externe &PAC7PJ
- . Fichier mouvement en entrée :
  - Mouvements utilisateurs  
PACDMB : fichier temporaire TMBDEXP
- . Fichiers de tri
- . Fichier en sortie :
  - Mouvements de mise à jour DUPT  
fichier temporaire TPACDMV (recline 250)

- . 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

Mise en bibliothèque source du résultat

EDITION DES MOUVEMENTS DE MISE A JOUR : PDS610

- . Fichiers permanents en entrée :
  - Fichier des données  
PACDDA : om externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des libellés d'erreur  
PACDDE : om externe \$NMTU.\$ROOT\$ROOTDE
- . Fichier en entrée :
  - Mouvements de mise à jour de DSMS  
PACDMV : fichier temporaire TPACDMV (resize 250)
- . 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

```

COMM '*****';
COMM '* D S M S      : EXTRACTION DE MOUVEMENTS DU      *';
COMM '*              FICHER JOURNAL (DJ)                *';
COMM '*              *                                    *';
COMM '*              DSUSDEXP                            *';
COMM '*              =====                            *';
COMM '*  SORTIE :                                         *';
COMM '*      LES MOUVEMENTS RESULTANTS SONT RANGES DANS LA *';
COMM '*      BIBLIOTHEQUE : $NMLI.$LIBSU                  *';
COMM '*****';
MVL  PAC7PJ='PACPJ20',USER='$USER',SIZEWK=1,
      CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      RFTU=&CTTU$CTTU,
      CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      RFBU=&CTBU$CTBU,
      CTDJN='FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
      RFDJ=&CTDJ$CTDJ,
      RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*DEXP,
      OF=(TMBDEXP,TEMPRY,&RFTM,END=PASS),
      OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP PDS600,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
      SZ 60;
      ASG PACDMB,TMBDEXP,TEMPRY,&RFTM;
      ASG PACDMV,TPACDMV,TEMPRY,&RFTM,END=PASS;
      DEF PACDMV,CISZ=4096,RECSZ=250,RECFORM=FB;
      ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF PACDDE,READLOCK=STAT;
      ASG PACDDA,$NMTU.$ROOT$FILEDA,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF PACDDA,READLOCK=STAT;
      ASG PAC7PJ,&PAC7PJ,&RFBU;
      ASG PACDRU,SYS.OUT;
      ASG PAC7EI,SYS.OUT;
      SWK WKDISK=(SZ=&SIZEWK,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,0;
STEP PDS610,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
      SZ 60;
      ASG PACDMV,TPACDMV,TEMPRY,&RFTM,END=PASS;
      ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF PACDDE,READLOCK=STAT;
      ASG PACDDA,$NMTU.$ROOT$FILEDA,&RFTU,
          ACCESS=READ,SHARE=MONITOR;
      DEF PACDDA,READLOCK=STAT;
      ASG PAC7EI,SYS.OUT;
      ASG PACDRU,SYS.OUT;
ESTP;
LMN  SL INFILE=(TPACDMV,TEMPRY,&RFTM),
      LIB=($NMLI.$LIBSU,&RFLI),
      COM='MV INFILE:MBDUPD_DEXP'&USER',INFORM=SARF,
          TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SW20,EQ,1;

```

EXTRACTION DU JOURNAL ARCHIVE VA PAC  
JCL DE LA PROCEDURE

(DEXP)

PAGE

104

10  
4

```
JUMP END;  
ERR:  
SEND ' DSUSDEXP - ABNORMAL END OF RUN (I/O ERROR) ' ;  
LET SEV 3 ;  
END:
```



VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
EXTRACTION D'ENTITES

(DEXT)

PAGE 105

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'Événements 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 demandées 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 ! Requetes sur Ameliorations !
!   !   ! LCQE ! Requetes sur Evenements !
!   !   ! LCQS ! Requetes sur Sites !
! 02 ! 04 ! LCRC ! Maquettes sur Ameliorations !
!   !   ! LCRE ! Maquettes sur Evenements !
!   !   ! LCRS ! Maquettes sur Sites !
! 12 ! 08 !uuuuuuu! Utilisateur proprietaire des requetes!
!   !   !   ! 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 :!
!      !      !      ! (99 requêtes maximum) !
!  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 :

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Extraction des entités : PDSEX
- . Mise en bibliothèque source du fichier résultat

#### PRISE EN COMPTE DES ENTREES : CREATE

-----  
Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDEXT  
(reclsize 80, Csize 2048)
- Fichier des données : PACDDA

#### EXTRACTIONS : PDSEX

- . Fichiers permanents en entrée :
  - Fichier des données  
DS80IA : \$NMTU.\$ROOT\$FILEDA
  - Fichier des éléments VA Pac  
PB80DC : \$NMTU.\$ROOT\$FILEDC
  - Fichier des libellés d'erreur  
DS80IE : \$NMTU.\$ROOT\$FILEDE
- . Fichier mouvements en entrée :
  - Demandes d'extraction  
fichier temporaire TMBDEXT
- . Fichiers de travail :
  - Requêtes  
TPACDKQ
  - Fichiers temporaires  
TPACDW0 TPACDW1 TPACDW2 TPACDW3 TPACDW4 TPACDW5 TPACDWI
- . Etats en sortie :

**EXTRACTION D'ENTITES**  
**DESCRIPTION DES ETAPES**

(DEXT)

PAGE

111

11  
3

- Compte-rendu d'enchaînement  
PACDIA
- Compte-rendu demandes extraction  
PACDRU
- . Fichiers de tri
- . Fichier en sortie :
  - Mouvements batch extraits  
fichier temporaire TPACDIM

## 11.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '* D S M S      : EXTRACTION D"ENTITES DSMS      *';
COMM '*                                           *';
COMM '*                DSUSDEXT                       *';
COMM '*                =====                       *';
COMM '*  SORTIE :                                       *';
COMM '*    LES MOUVEMENTS RESULTANTS SONT RANGES DANS LA *';
COMM '*    BIBLIOTHEQUE : $NMLI.$LIBSU                 *';
COMM '*****';
MVL  USER=' $USER' , SIZEWK=1 ,
      CTTUN= ' FILESTAT=UNCAT , DVC=$DVTU , MD=$MDTU ' ,
      RFTU=&CTTU$CTTU ,
      CTBSN= ' FILESTAT=UNCAT , DVC=$DVBS , MD=$MDBS ' ,
      RFBS=&CTBS$CTBS ,
      CTLIN= ' FILESTAT=UNCAT , DVC=$DVLI , MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      CTBUN= ' FILESTAT=UNCAT , DVC=$DVBU , MD=$MDBU ' ,
      RFBU=&CTBU$CTBU ,
      CTDJN= ' FILESTAT=UNCAT , DVC=$DVDJ , MD=$MDDJ ' ,
      RFDJ=&CTDJ$CTDJ ,
      RFTM= ' DVC=$DVTM , MD=$MDTM ' ,
      PGE= ' E ' , PGLNG=&PG$LANG;
CR   IF=*DEXT ,
      OF=( TMBDEXT , TEMPRY , &RFTM , END=PASS ) ,
      OUTDEF=( CISZ=2048 , RECSZ=80 , RECFORM=FB );
STEP PDSEX&PGLNG , FILE=( $NMLI . $LIBLM , &RFLI ) , DUMP=DATA;
SZ   100;
ASG  PACDMB , TMBDEXT , TEMPRY , &RFTM;
ASG  PACDIM , TPACDIM , TEMPRY , &RFTM , END=PASS;
DEF  PACDIM , CISZ=4096 , RECSZ=250 , RECFORM=FB;
ASG  PB80DC , $NMTU . $ROOT$FILEDC , &RFTU ,
      ACCESS=READ , SHARE=MONITOR;
DEF  PB80DC , READLOCK=STAT;
ASG  DS80IA , $NMTU . $ROOT$FILEDA , &RFTU ,
      ACCESS=READ , SHARE=MONITOR;
DEF  DS80IA , READLOCK=STAT;
ASG  DS80IE , $NMTU . $ROOT$ROOTDE , &RFTU ,
      ACCESS=READ , SHARE=MONITOR;
DEF  DS80IE , READLOCK=STAT;
ASG  PACDKQ , TPACDKQ , TEMPRY , &RFTM , END=PASS;
ASG  PACDWI , TPACDWI , TEMPRY , &RFTM , END=PASS;
ALC  PACDWI , SZ=&SIZEWK , UNIT=CYL;
DEF  PACDWI , CISIZE=$CISEQ , NBBUF=1;
ASG  PACDW0 , TPACDW0 , TEMPRY , &RFTM , END=PASS;
ALC  PACDW0 , SZ=&SIZEWK , UNIT=CYL;
DEF  PACDW0 , CISIZE=$CISEQ , NBBUF=1;
ASG  PACDW1 , TPACDW1 , TEMPRY , &RFTM , END=PASS;
ALC  PACDW1 , SZ=&SIZEWK , UNIT=CYL;
DEF  PACDW1 , CISIZE=$CISEQ , NBBUF=1;
ASG  PACDW2 , TPACDW2 , TEMPRY , &RFTM , END=PASS;
ALC  PACDW2 , SZ=&SIZEWK , UNIT=CYL;
DEF  PACDW2 , CISIZE=$CISEQ , NBBUF=1;
ASG  PACDW3 , TPACDW3 , TEMPRY , &RFTM , END=PASS;
ALC  PACDW3 , SZ=&SIZEWK , UNIT=CYL;
DEF  PACDW3 , CISIZE=$CISEQ , NBBUF=1;
ASG  PACDW4 , TPACDW4 , TEMPRY , &RFTM , END=PASS;
ALC  PACDW4 , SZ=&SIZEWK , UNIT=CYL;
DEF  PACDW4 , CISIZE=$CISEQ , NBBUF=1;
ASG  PACDW5 , TPACDW5 , TEMPRY , &RFTM , END=PASS;
ALC  PACDW5 , SZ=&SIZEWK , UNIT=CYL;

```



```
DEF PACDW5,CISIZE=$CISEQ,NBBUF=1;
ASG PACDRU,SYS.OUT;
ASG PACDIA,SYS.OUT;
ASG PAC7EI,SYS.OUT;
SWK WKDISK=(SZ=&SIZEWK,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP ERR,SW30,EQ,1;
LMN SL INFILE=(TPACDIM,TEMPRY,&RFTM),
LIB=( $NMLI.$LIBSU,&RFLI),
COM='MV INFILE:MBDUPD_DEXT'&USER',INFORM=SARF,
TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' DSUSDEXT - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:
```

EXTRACTION D'ENTITES  
JCL DE LA PROCEDURE

(DEXT)

PAGE

114

11  
4

VISUALAGE PACBASE - MANUEL D'EXPLOITATION	PAGE	115
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

#### NOTATIONS UTILISEES

```
-----  
! NOTATION ! SIGNIFICATION <EXEMPLE>!  
!-----!  
! &SIZEWK ! réservé au tri interne 01 !  
! &USER ! nom de l'utilisateur DSMS IBM !  
!-----!
```

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Extraction-formatage mouvements : PDSXTH

#### PRISE EN COMPTE DES ENTREES : CREATE

-----  
Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

#### 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  
DS80IA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des libellés d'erreur  
DS80IE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Fichier mouvements en entrée :
  - Contrôle de l'utilisateur  
fichier temporaire TMBDEXH
- . Fichier en sortie :
  - Tables extraites  
PACDMV

EXTRACTION DE TABLES POUR LISTES EXTERNES (DEXH)  
DESCRIPTION DES ETAPES

PAGE

119

12  
3

. Etat en sortie :  
- Compte-rendu d'extraction  
PACDRH

. Fichiers de tri

Mise en bibliothèque source du résultat

## 12.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '* D S M S      : EXTRACTION DES TABLES DSMS      *';
COMM '*                                                    *';
COMM '*              DSUSDEXH                            *';
COMM '*              =====                            *';
COMM '* SORTIE :                                          *';
COMM '* LES MOUVEMENTS RESULTANTS SONT RANGES DANS LA   *';
COMM '* BIBLIOTHEQUE : $NMLI.$LIBSU                      *';
COMM '*****';
MVL  USER=' $USER' ,SIZEWK=1 ,
      CTTUN= ' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU ' ,
      RFTU=&CTTU$CTTU ,
      CTBSN= ' FILESTAT=UNCAT ,DVC=$DVBS ,MD=$MDBS ' ,
      RFBS=&CTBS$CTBS ,
      CTLIN= ' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      CTBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
      RFBU=&CTBU$CTBU ,
      CTDJN= ' FILESTAT=UNCAT ,DVC=$DVDJ ,MD=$MDDJ ' ,
      RFDJ=&CTDJ$CTDJ ,
      RFTM= ' DVC=$DVTM ,MD=$MDTM ' ;
CR   IF= *DEXH ,
      OF= ( TMBDEXH ,TEMPRY ,&RFTM ,END=PASS ) ,
      OUTDEF= ( CISZ=2048 ,RECSZ=80 ,RECFORM=FB ) ;
STEP PDSXTH ,FILE= ( $NMLI . $LIBLM ,&RFLI ) ,DUMP=DATA ;
      SZ 60 ;
      ASG PACDMB ,TMBDEXH ,TEMPRY ,&RFTM ;
      ASG PACDMV ,TPACDMV ,TEMPRY ,&RFTM ,END=PASS ;
      DEF PACDMV ,CISZ=2048 ,RECSZ=100 ,RECFORM=FB ;
      ASG DS80IE , $NMTU . $ROOT$ROOTDE ,&RFTU ,
          ACCESS=READ ,SHARE=MONITOR ;
      DEF DS80IE ,READLOCK=STAT ;
      ASG DS80IA , $NMTU . $ROOT$FILEDA ,&RFTU ,
          ACCESS=READ ,SHARE=MONITOR ;
      DEF DS80IA ,READLOCK=STAT ;
      ASG PACDRH ,SYS .OUT ;
      ASG PAC7EI ,SYS .OUT ;
      SWK WKDISK= ( SZ=&SIZEWK ,&RFTM ) ;
ESTP ;
JUMP ERR ,SW20 ,EQ ,1 ;
LMN  SL INFILE= ( TPACDMV ,TEMPRY ,&RFTM ) ,
      LIB= ( $NMLI . $LIBSU ,&RFLI ) ,
      COM= ' MV INFILE:MBPAW_DEXH '&USER' ,INFORM=SARF ,
          TYPE=DAT ,NUMBER= ( 1 ,1 ) ,REPLACE ; ' ;
JUMP ERR ,SW20 ,EQ ,1 ;
JUMP END ;
ERR :
SEND ' DSUSDEXH - ABNORMAL END OF RUN ( I/O ERROR ) ' ;
LET  SEV 3 ;
END :

```



VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
MISE A JOUR BATCH

(DUPT)

PAGE 121

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 utili-!
!    !    !        !   sateur/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 utili-!
!    !    !        !   sateur/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 nu- !
!    !    !        !   méro 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 ex-	!
!	!	!	! ternes d'événements et d'améliora-	!
!	!	!	! tions, 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 impri-	!
!	!	!	! mée 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

##### NOTATIONS UTILISEES

```
+-----+-----+-----+
! NOTATION ! SIGNIFICATION          <EXEMPLE>!
+-----+-----+-----+
! &USER    ! CODE UTILISATEUR       <$USER>!
+-----+-----+-----+
```

##### LANCEMENT DE LA PROCEDURE

Cette procédure peut être lancée de deux façons différentes:

EJ DSIVDUPT, pour prendre comme entrée utilisateur les  
mouvements saisis dans la carte \$INPUT de DSIVDUPT

EJ DSIVDUPT VL=(DEXP, DEXT ou DXBJ) pour prendre comme  
entrée utilisateur les mouvements issus de DEXP,  
DEXT ou DXBJ.

Nom externe : \$NMTU.\$ROOT\$ROOTDE

##### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

```
.Prise en compte des entrées          : CREATE
.Contrôle de l'intégrité de la base   : PDSBAS
.Mise à jour                          : PDSUP0
```

##### PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir des  
cartes DATA définies dans la procédure d'appel DSIVDUPT  
ou d'un fichier issu des procédures d'extraction.

```
.Fichier en entrée                    : $INPUT
                                       ou MBDUPD_&1&USER
.Fichier en sortie                    : TMBDUPT
```

##### VERIFICATION INTEGRITE DE LA BASE : PDSBAS

```
. Fichier permanent en entrée :
- Fichier des données          : PACDDA
  Nom externe : $NMTU.$ROOT$FILEDA
- Fichier des libellés d'erreurs : PACDDE

. Etats en sortie :
- Compte rendu de validité     : PACDRS
```

- Compte rendu d'anomalie : PAC7EI
- . Code retour : Switch-30
  - 0 : la base est valide.
  - 1 : la base n'est pas valide. Dans ce cas, aucune autre étape de la procédure ne sera exécutée.

MISE A JOUR DE LA BASE DSMS : PDSUP0

- . Fichiers permanents en entrée-sortie :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des éléments VisualAge Pacbase  
PACDDC : Nom externe \$NMTU.\$ROOT\$FILEDC
  - Fichier des références croisées  
PACDDX : Nom externe \$NMTU.\$ROOT\$FILEDX
- . Fichier permanent en entrée :
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Fichier mouvement en entrée :
  - Mouvements de mise jour issus de DEXP  
PACDIM : Nom externe TMBDUPT
- . Fichier en sortie :
  - Journal  
PACDDJ : Nom externe \$NMTU.\$ROOT\$FILEDJ
- . Etat en sortie :
  - Compte-rendu de mise à jour  
PACDRP et PACDEI pour les anomalies
- . 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

```

COMM '*****';
COMM '* D S M S      : MISE A JOUR DE LA BASE DSMS      *';
COMM '*                                           *';
COMM '*              DSUSDUPT                          *';
COMM '*              =====                          *';
COMM '*****';
MVL  USER=' $USER ',
      CTTUN=' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU ',
      RFTU=&CTTU$CTTU ,
      CTBSN=' FILESTAT=UNCAT ,DVC=$DVBS ,MD=$MDBS ',
      RFBS=&CTBS$CTBS ,
      CTLIN=' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ',
      RFLI=&CTLI$CTLI ,
      CTBUN=' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ',
      RFBU=&CTBU$CTBU ,
      CTDJN=' FILESTAT=UNCAT ,DVC=$DVDJ ,MD=$MDDJ ',
      RFDJ=&CTDJ$CTDJ ,
      RFTM=' DVC=$DVTM ,MD=$MDTM ' ;
JUMP CR&1 ;
CRJCL:
CR   IF=*DUPT ,
      OF=( TMBDUPT ,TEMPRY ,&RFTM ,END=PASS ) ,
      OUTDEF=( CISZ=2048 ,RECSZ=250 ,RECFORM=FB ) ;
JUMP CREND ;
CRDEXP:CRDEXT:CRDXBJ:
CR   IF=( $NMLI . $LIBSU ,&RFLI ,SUBFILE=MBDUPD_&1&USER ) ,
      OF=( TMBDUPT ,TEMPRY ,&RFTM ,END=PASS ) ,
      OUTDEF=( CISZ=2048 ,RECSZ=250 ,RECFORM=FB ) ,
      COMFILE=*SELREC ,START=2 ;
CREND:
COMM '*** PDSBAS ***' ;
STEP PDSBAS ,FILE=( $NMLI . $LIBLM ,&RFLI ) ,DUMP=DATA ;
      SZ 30 ;
      ASG PACDDA , $NMTU . $ROOT$FILEDA , &RFTU ;
      ASG PACDDE , $NMTU . $ROOT$ROOTDE , &RFTU ;
      ASG PACDRS , SYS . OUT ;
      ASG PAC7EI , SYS . OUT ;
ESTP ;
JUMP ERR , SW20 , EQ , 1 ;
JUMP END , SW30 , EQ , 1 ;
COMM '*** PDSUP0 ***' ;
STEP PDSUP0 ,FILE=( $NMLI . $LIBLM ,&RFLI ) ,DUMP=DATA ;
      SZ 100 ;
      ASG DS80IA , $NMTU . $ROOT$FILEDA , &RFTU ;
      ASG DS80DJ , $NMDJ . $ROOT$FILEDJ , &RFDJ ;
      ASG DS80IX , $NMTU . $ROOT$FILEDX , &RFTU ;
      ASG PB80DC , $NMTU . $ROOT$FILEDC , &RFTU ;
      ASG DS80IE , $NMTU . $ROOT$ROOTDE , &RFTU ,
          ACCESS=READ , SHARE=MONITOR ;
      DEF DS80IE , READLOCK=STAT ;
      ASG PACDIM , TMBDUPT , TEMPRY , &RFTM ;
      ASG PACDRP , SYS . OUT ;
      ASG PAC7EI , SYS . OUT ;
ESTP ;
JUMP ERR , SW20 , EQ , 1 ;
JUMP END ;
ERR:
SEND ' DSUSDUPT - ABNORMAL END OF RUN ( I/O ERROR ) ' ;
LET SEV 3 ;
END:

```

MISE A JOUR BATCH  
JCL DE LA PROCEDURE

(DUPT)

PAGE

132

13  
4

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
INITIALISATION

(DINI)

PAGE 133

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 :

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Sauvegarde initiale de la base : PDSINI
- . Rotation des générations de BB : SHIFT

#### PRISE EN COMPTE DES ENTREES : CREATE

-----

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDINI  
(recline 80, Csize 2048)

#### SAUVEGARDE INITIALE DE LA BASE : PDSINI

- . Fichiers mouvement en entrée :
  - Mouvement d'initialisation
  - PACDDB : fichier temporaire TMBDINI
- . Fichier permanent en entrée :
  - Libellés d'erreur
  - PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
- . Fichier en sortie :
  - Image séquentielle des fichiers
  - PACDBB : Nom externe \$NMBU.\$ROOT\$FILEEBB/G+1
- . Etat en sortie :
  - Compte rendu de sauvegarde
  - PACDRU

## 14.4. JCL DE LA PROCEDURE

```
COMM '*****';
COMM '* DSMS : INITIALISATION DES FICHIERS *';
COMM '* *';
COMM '* DSEXDINI *';
COMM '* ===== *';
COMM '* *';
COMM '*****';
MVL CTTUN= ' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU ' ,
RFTU=&CTTU$CTTU ,
CTBSN= ' FILESTAT=UNCAT ,DVC=$DVBS ,MD=$MDBS ' ,
RFBS=&CTBS$CTBS ,
CTLIN= ' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ' ,
RFLI=&CTLI$CTLI ,
CTBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
RFBU=&CTBU$CTBU ,
CTDJN= ' FILESTAT=UNCAT ,DVC=$DVDJ ,MD=$MDDJ ' ,
RFDJ=&CTDJ$CTDJ ,
RFTM= ' DVC=$DVTM ,MD=$MDTM ' ;
CR IF=*DINI ,
OF= ( TMBDINI , &RFTM , TEMPRY , END=PASS ) ,
OUTDEF= ( CISZ=2048 , RECSZ=80 , RECFORM=FB ) ;
STEP PDSINI , FILE= ( $NMLI . $LIBLM , &RFLI ) , DUMP=DATA ;
SZ 40 ;
ASG PACDMB , TMBDINI , TEMPRY , &RFTM ;
ASG PACDBB , $NMBU . $ROOT$FILEBB / G+1 , &RFBU ;
ASG PACDDE , $NMTU . $ROOT$ROOTDE , &RFTU ,
ACCESS=READ , SHARE=MONITOR ;
DEF PACDDE , READLOCK=STAT ;
ASG PACDRU , SYS . OUT ;
ASG PAC7EI , SYS . OUT ;
ESTP ;
JUMP ERR , SW20 , EQ , 1 ;
SHIFT $NMBU . $ROOT$FILEBB ;
JUMP END ;
ERR :
SEND ' DSEXDINI - ABNORMAL END OF RUN ( I/O ERROR ) ' ;
LET SEV 3 ;
END :
```

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
EXTRACTION DU JOURNAL POUR MISE A JOUR (DXBJ)

PAGE 139

15

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

## 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.	!
!	!	! '1'	! 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

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Extraction du journal : PDS700
- . Mise en bibliothèque source du résultat

PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDXBJ  
(reclsize 80, Csize 2048)

EXTRACTION ET FORMATAGE DES MOUVEMENTS : PDS700

- . Fichiers permanents en entrée :
  - Fichier des données  
PACDDA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des libellés d'erreur  
PACDDE : Nom externe \$NMTU.\$ROOT\$ROOTDE
  - Journal DSMS archivé  
PACDBJ : Nom externe \$NMBU.\$ROOT\$FILEBJ
- . Fichier mouvement en entrée :
  - Mouvements utilisateurs  
PACDMB : Fichier temporaire TMBDXBJ
- . Fichier en sortie :
  - Fichier mouvements MAJ pour DUPT  
PACDIM : Fichier temporaire TPACDIM (reclsize 250)
- . 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

```

COMM '*****';
COMM '* D S M S      : EXTRACTION DU JOURNAL ARCHIVE      *';
COMM '*                                                    *';
COMM '*              DSUSDXBJ                               *';
COMM '*              =====                               *';
COMM '*  SORTIE :                                           *';
COMM '*    LES MOUVEMENTS RESULTANTS SONT RANGES DANS LA  *';
COMM '*    BIBLIOTHEQUE : $NMLI.$LIBSU                      *';
COMM '*****';
MVL  USER=' $USER ', SIZEWK=1,
      CTTUN=' FILESTAT=UNCAT, DVC=$DVTU, MD=$MDTU ',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT, DVC=$DVBS, MD=$MDBS ',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT, DVC=$DVLI, MD=$MDLI ',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT, DVC=$DVBU, MD=$MDBU ',
      RFBU=&CTBU$CTBU,
      CTDJN=' FILESTAT=UNCAT, DVC=$DVDJ, MD=$MDDJ ',
      RFDJ=&CTDJ$CTDJ,
      RFTM=' DVC=$DVTM, MD=$MDTM ';
CR   IF=*DXBJ,
      OF=( TMBDXBJ, TEMPRY, &RFTM, END=PASS ),
      OUTDEF=( Cisz=2048, RECSZ=80, RECFORM=FB );
STEP PDS700, FILE=( $NMLI.$LIBLM, &RFLI ), DUMP=DATA;
      SZ 100;
      ASG PACDMB, TMBDXBJ, TEMPRY, &RFTM;
      ASG PACDIM, TPACDIM, TEMPRY, &RFTM, END=PASS;
      DEF PACDIM, Cisz=4096, RECSZ=250, RECFORM=FB;
      ASG DS80IA, $NMTU.$ROOT$FILEDA, &RFTU,
          ACCESS=READ, SHARE=MONITOR;
      DEF DS80IA, READLOCK=STAT;
      ASG DS80IE, $NMTU.$ROOT$ROOTDE, &RFTU,
          ACCESS=READ, SHARE=MONITOR;
      DEF DS80IE, READLOCK=STAT;
      ASG PACDBJ, $NMBU.$ROOT$FILEBJ, &RFBU;
      ASG PACDRK, SYS.OUT;
      ASG PACDSK, SYS.OUT;
      ASG PAC7EI, SYS.OUT;
      SWK WKDISK=( SZ=&SIZEWK, &RFTM );
ESTP;
JUMP ERR, SW20, EQ, 1;
JUMP END, SW30, EQ, 1;
LMN  SL INFILE=( TPACDIM, TEMPRY, &RFTM ),
      LIB=( $NMLI.$LIBSU, &RFLI ),
      COM=' MV INFILE:MBDUPD_DXBJ'&USER', INFORM=SARF,
          TYPE=DAT, NUMBER=( 1, 1 ), REPLACE; ';
JUMP ERR, SW20, EQ, 1;
JUMP END;
ERR:
SEND ' DSUSDXBJ - ABNORMAL END OF RUN ( I/O ERROR ) ';
LET  SEV 3;
END:

```



VISUALAGE PACBASE - MANUEL D'EXPLOITATION	PAGE	145
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 !
+-----+
```

## 16.3. 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 :

#### NOTATIONS UTILISEES

```
-----  
! NOTATION ! SIGNIFICATION <EXEMPLE>!  
!-----!  
! &SIZEWK ! taille des fichiers intermédiaires 01 !  
-----
```

#### DESCRIPTION DES ETAPES

-----

Cette procédure comprend les étapes suivantes :

- . Prise en compte des entrées : CREATE
- . Modifications : PDSMS
- . Rotation des sauvegardes : SHIFT

#### PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

- . Fichier en entrée : \$INPUT
- . Fichier en sortie (TEMPRY) : TMBDREN  
(recline 80, Csize 2048)

#### MODIFICATIONS : PDSMS

- . Fichiers permanents en entrée :
  - Fichier des données  
DS80IA : Nom externe \$NMTU.\$ROOT\$FILEDA
  - Fichier des libellés d'erreur  
DS80IE : Nom externe \$NMTU.\$ROOT\$ROOTDE
  - Fichier des références croisées  
DS80IX : Nom externe \$NMTU.\$ROOT\$FILEDX
  - Sauvegarde DSMS  
PACDBB : Nom externe \$NMBU.\$ROOT\$FILEBB

- Archivage DSMS  
PACDBJ : Nom externe \$NMBU.\$ROOT\$FILEBJ
- . Fichier en entrée :
  - Requêtes utilisateur  
PACDMB : Temporaire TMBDREN
- . Fichiers de travail :
  - Demandes de modifications  
PACDW0
  - Sauvegarde partielle triée  
PACDW1
  - Sauvegarde partielle non triée  
PACDW2
- . Fichiers en sortie :
  - Sauvegarde modifiée  
PACDB3 : Nom externe \$NMBU.\$ROOT\$FILEBB/g+1
  - Archive modifiée  
PACDJB : Nom externe \$NMBU.\$ROOT\$FILEBJ/g+1
- . 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

## 16.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '*          DSMS      : MODIF DE CODES ET MOTS CLES      *';
COMM '*                                                    *';
COMM '*                  DSEXDRN                            *';
COMM '*                  =====                            *';
COMM '*                                                    *';
COMM '******';
MVL  SIZEWK=2,
      CTTUN=' FILESTAT=UNCAT, DVC=$DVTU, MD=$MDTU' ,
      RFTU=&CTTU$CTTU ,
      CTBSN=' FILESTAT=UNCAT, DVC=$DVBS, MD=$MDBS' ,
      RFBS=&CTBS$CTBS ,
      CTTLN=' FILESTAT=UNCAT, DVC=$DVLI, MD=$MDLI' ,
      RFLI=&CTLI$CTLI ,
      CTBUN=' FILESTAT=UNCAT, DVC=$DVBU, MD=$MDBU' ,
      RFBU=&CTBU$CTBU ,
      CTDJN=' FILESTAT=UNCAT, DVC=$DVDJ, MD=$MDDJ' ,
      RFDJ=&CTDJ$CTDJ ,
      RFTM=' DVC=$DVTM, MD=$MDTM' ,
      PGE=' E' , PGLNG=&PGL$LANG;
CR   IF=*DREN,
      OF=( TMBDREN, &RFTM, TEMPRY, END=PASS ) ,
      OUTDEF=( CISZ=2048, RECSZ=80, RECFORM=FB );
STEP PDSMS&PGLNG, FILE=( $NMLI . $LIBLM, &RFLI ) , DUMP=DATA;
SZ   100;
ASG  PACDMB, TMBDREN, TEMPRY, &RFTM, END=PASS;
ASG  DS80IA, $NMTU. $ROOT$FILEDA, &RFTU,
      ACC=READ, SHARE=MONITOR;
DEF  DS80IA, READLOCK=STAT;
ASG  DS80IE, $NMTU. $ROOT$ROOTDE, &RFTU,
      ACC=READ, SHARE=MONITOR;
DEF  DS80IE, READLOCK=STAT;
ASG  DS80IX, $NMTU. $ROOT$FILEDX, &RFTU,
      ACC=READ, SHARE=MONITOR;
DEF  DS80IX, READLOCK=STAT;
ASG  PACDBB, $NMBU. $ROOT$FILEBB, &RFBU;
ASG  PACDBJ, $NMBU. $ROOT$FILEBJ, &RFBU;
ASG  PACDW0, TPACDW0, TEMPRY, &RFTM;
ALC  PACDW0, SZ=&SIZEWK, INCRSZ=1, UNIT=CYL;
DEF  PACDW0, CISIZE=$CISEQ, NBBUF=1;
ASG  PACDW1, TPACDW1, TEMPRY, &RFTM;
ALC  PACDW1, SZ=&SIZEWK, INCRSZ=1, UNIT=CYL;
DEF  PACDW1, CISIZE=$CISEQ, NBBUF=1;
ASG  PACDW2, TPACDW2, TEMPRY, &RFTM;
ALC  PACDW2, SZ=&SIZEWK, INCRSZ=1, UNIT=CYL;
DEF  PACDW2, CISIZE=$CISEQ, NBBUF=1;
ASG  PACDB3, $NMBU. $ROOT$FILEBB/G+1, &RFBU;
ASG  PACDJB, $NMBU. $ROOT$FILEBJ/G+1, &RFBU;
ASG  PACDIA, SYS. OUT;
ASG  PACDIK, SYS. OUT;
ASG  PACDJK, SYS. OUT;
ASG  PACDIS, SYS. OUT;
ASG  PACDKK, SYS. OUT;
ASG  PACDLK, SYS. OUT;
ASG  PAC7EI, SYS. OUT;
SWK  WKDISK=( SZ=&SIZEWK, &RFTM );
ESTP;
JUMP ERR, SW20, EQ, 1;
JUMP ERR, SW30, EQ, 1;
COMM '*** CTL FOR SHIFT ***';

```



```
FILLIST INFILE=$NMBU.$ROOT$FILEBB/G+1 SPACE PRTPFILE=DUMMY;  
JUMP SHBBNO SEV GE 3;  
JUMP SHBBNO SW21 EQ 1;  
JUMP SHBBNO SW23 EQ 1;  
SHIFT $NMBU.$ROOT$FILEBB;  
SHBBNO:  
FILLIST INFILE=$NMBU.$ROOT$FILEBJ/G+1 SPACE PRTPFILE=DUMMY;  
JUMP SHBJNO SEV GE 3;  
JUMP SHBJNO SW22 EQ 1;  
JUMP SHBJNO SW23 EQ 1;  
SHIFT $NMBU.$ROOT$FILEBJ;  
SHBJNO:  
JUMP END;  
ERR:  
SEND ' DSEXDREN - ABNORMAL END OF RUN (I/O ERROR) '  
LET SEV 3;  
END:
```

MODIFICATION DE CODES ET DE MOTS CLES  
JCL DE LA PROCEDURE

(DREN)

PAGE

154

16  
4

VISUALAGE PACBASE - MANUEL D'EXPLOITATION	PAGE	155
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 :

#### NOTATIONS UTILISEES

```
+-----+-----+
! NOTATION ! SIGNIFICATION                <EXEMPLE>!
+-----+-----+
! &USER    ! CODE UTILISATEUR             <$USER>!
+-----+-----+
```

#### DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

```
.Prise en compte des entrées      : CREATE
.Pré-processeur                    : DAFD10
.Mise en bibliothèque source du résultat
```

#### PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un programme généré utilisateur contenant des ordres SQL.

```
.Fichier en entrée                  : &l
                                     (programme généré)
.Fichier en sortie                   : TMBDPDF
```

#### PRE-PROCESSEUR DU GENERE : DAFD10

```
. Fichiers permanents en entrée :
- Fichier des données
DS80IA : Nom externe $NMTU.$ROOT$FILEDA
- Fichier des libellés d'erreur
DS80IE : Nom externe $NMTU.$ROOT$ROOTDE

. Fichier en entrée :
- Programmes générés
DAF80 : fichier temporaire TMBDPDF

. Fichiers en sortie :
- Programmes générés à compiler
COB80 : fichier temporaire TMBDPDF

. Etats en sortie :
```

PRE-PROCESSEUR DAF PROGRAMMES GENERES	(DPDF)	PAGE	160
DESCRIPTION DES ETAPES			17
			3

- 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 (tmbdpdf) peut être remplacé par son envoi dans l'Internal Reader de la machine.



## 17.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '*          DSUSDPDF          *';
COMM '*          =====          *';
COMM '*          *          *';
COMM '*          PRE-PROCESSING D'UN SOURCE VA PAC DAF          *';
COMM '*          *          *';
COMM '*****';
MVL  DAF,
      CTTUN= ' FILESTAT=UNCAT, DVC=$DVTU, MD=$MDTU ' ,
      RFTU=&CTTU$CTTU ,
      CTBSN= ' FILESTAT=UNCAT, DVC=$DVBS, MD=$MDBS ' ,
      RFBS=&CTBS$CTBS ,
      CTLIN= ' FILESTAT=UNCAT, DVC=$DVLI, MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      CTDJN= ' FILESTAT=UNCAT, DVC=$DVDJ, MD=$MDDJ ' ,
      RFDJ=&CTDJ$CTDJ ,
      RFTM= ' DVC=$DVTM, MD=$MDTM ' ;
MVT:
CR   IF=*&1,
      OF=( TMBDPDF, TEMPRY, &RFTM, END=PASS ),
      OUTDEF=( CISZ=2048, RECSZ=80 );
STEP DAFD10, FILE=( $NMLI.$LIBLM, &RFLI ), DUMP=DATA;
SZ   130;
ASG  DS80IE, $NMTU.$ROOT$ROOTDE, &RFTU,
      ACC=READ, SHARE=MONITOR;
DEF  DS80IE, NBBUF=2, READLOCK=STAT;
ASG  DS80IA, $NMTU.$ROOT$FILEDA, &RFTU,
      ACC=READ, SHARE=MONITOR;
DEF  DS80IA, READLOCK=STAT;
ASG  DAF80, TMBDPDF, TEMPRY, &RFTM, END=PASS;
DEF  DAF80, NBBUF=1;
ASG  COB80, &1, TEMPRY, &RFTM, END=PASS;
DEF  COB80, NBBUF=1;
ASG  DAFREP, SYS.OUT;
ESTP;
LMN  SL INFILE=( &1, TEMPRY, &RFTM, END=PASS ),
      LIB=( $NMLI.$LIBSU, &RFLI ),
      COM= ' MV INFILE: '&1', INFORM=SARF,
           TYPE=DAT, NUMBER=( 1, 1 ), REPLACE; ';
JUMP END, SW30, EQ, 0;
JUMP ERR, SW20, EQ, 1;
JUMP END;
ERR:
SEND ' DSUSDPDF - ABNORMAL END OF RUN (I/O ERROR) ' ;
LET  SEV 3;
END:

```

PRE-PROCESSEUR DAF PROGRAMMES GENERES  
JCL DE LA PROCEDURE

(DPDF)

PAGE

162

17  
4

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
I N S T A L L A T I O N

PAGE 163

18

## 18. INSTALLATION

## 18.1. 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 bande d'implantation et son déroulement est décrit dans le chapitre ci-dessous.

Avant de procéder à l'implantation, l'utilisateur doit avoir pris connaissance des caractéristiques techniques du système DSMS 2.0 décrites dans le chapitre environnement du présent manuel afin de prévoir l'environnement nécessaire au bon déroulement de l'implantation (place sur disques, définition du TDS et des utilisateurs de ce TDS au niveau du catalogue, etc...).

## PREPARATION DE L'INSTALLATION

Définition d'une bibliothèque de JCL, chargement de cette bibliothèque à partir d'une sauvegarde contenue sur la bande et paramétrage de ce JCL.

- . Sauvegarde de la bande d'implantation,
- . Allocation et déchargement d'une bibliothèque contenant le JCL complet d'implantation et d'exploitation de DSMS.
- . Adaptation de ce JCL aux particularités du site.

## IMPLANTATION

- . Préallocation des fichiers du système,
- . Préparation du TDS,
- . Implantation des fichiers,
- . Génération du TDS,
- . Link-edit des programmes,
- . Implantation d'une base DSMS de tests.

## TESTS

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

## 18.2. BANDE D'IMPLANTATION

### BANDE D'IMPLANTATION

La bande d'implantation (6250 BPI, labels standards)  
contient les fichiers suivants :

!Rang!	Label	! Contenu	!
! 1 !	! SVF.JCL	! JCL squelette d'implantation et	!
! !	! !	! d'exploitation français	!
! 2 !	! SVE.JCL	! JCL squelette d'implantation et	!
! !	! !	! d'exploitation anglais	!
! !	! !	! !	!
! 3 !	! SV.SL	! Sources TDS et TPRs	!
! !	! !	! !	!
! 4 !	! SV.CUB	! Compil-units batch	!
! !	! !	! !	!
! 5 !	! SV.CUT	! Compil-units TPRs	!
! !	! !	! !	!
! 6 !	! SV.DE0	! Fichier systeme des libelles d'erreur!	!
! !	! !	! et de documentation	!
! !	! !	! !	!
! 7 !	! SV.BB	! Fichier sauvegarde base DSMS	!
! !	! !	! !	!
! 8 !	! SV.SOURCE	! Sources des sous-programmes utilisat.!	!
! !	! !	! !	!
! 9 !	! SVF.ABOUT	! Caractéristiques de la version en	!
! !	! !	! français	!
! 10 !	! SVE.ABOUT	! Caractéristiques de la version en	!
! !	! !	! anglais	!

## 18.3. INSTALLATION DU JCL

### INSTALLATION DU JCL COMPLET

Cette installation se déroule en trois étapes:

- 1- Allocation d'une librairie de JCL '\$NMLI.\$LIBJCL' par BLIB (members=70,sz=1). Le nom de cette librairie doit être conforme aux valeurs prises par les paramètres constituant son nom et qui seront indiquées dans le membre 'DSZZVALS' lors de l'adaptation du JCL.
- 2- Chargement de cette librairie à partir du premier fichier de la bande (SVF.JCL) par LIBMAINT ou du second fichier 'SVE.JCL' pour une version Anglaise.
- 3- Adaptation de ce JCL aux particularités du site. Elle s'effectue par la modification à l'éditeur du membre 'DSZZVALS' de cette librairie (la valeur prise par défaut pour chacun des paramètres est substituée par sa valeur prise sur le site).

Ensuite, il est nécessaire de lancer la procédure 'DSZZEXEC' qui prépare le JCL de remplacement des paramètres (EXEC DSZZEXEC VL=DSZZJCL BRIEF;)

Exécuter ensuite ce JCL (SUBMIT DSZZJCL;).

### NOTES :

Lors de l'exécution de 'DSZZEXEC', des messages d'erreur du type 'SUBSTITUTION FAILED' apparaissent mais ils ne perturbent pas l'exécution.

Le TDS de DSMS est normalement linké dans la bibliothèque de Load-modules Batch de DSMS. Il est donc important de donner à \$NMTD une valeur différente des load-modules standard de DSMS, (voir la liste des load-modules batch au chapitre "LES COMPOSANTS DE DSMS").

TABLEAU DES MEMBRES DE JCL DE VA Pac

Tous ces membres sont contenus dans la librairie  
 '\$NMLI.\$LIBJCL'.

PROCEDURES D'EXPLOITATION

-----

! Membre	! Contenu	!Nature!
! DSEXDARC	! Archivage du journal	! JCL !
! DSEXDRST	! Restauration de la base	! JCL !
! DSEXDSAV	! Sauvegarde de la base	! JCL !
! DSEXPDSL	! Membre technologique pour transformer	! JCL !
!	! un membre IOF en UFAS 80 caractères	! !
! DSEXDREO	! Réorganisation de la base	! JCL !
! DSEXDINI	! Initialisation de la base	! JCL !
! DSEXJOBL	! Lanceur de JOBS pour écran GP	! JCL !
! DSEXTDDS	! Lancement du TDS DSMS	! JCL !
! DSEXDREN	! Changement de codes d'entités	! JCL !
! DSEXDLVB	! Remplacement de low value par blanc	! JCL !

PROCEDURES UTILISATEUR

-----

! Membre	! Contenu	!Nature!
! DSUSDEXP	! Extraction du journal VA Pac	! JCL !
! DSUSDEXQ	! Extr. du journal VA Pac anterieur à 2.0	! JCL !
! DSUSDEXT	! Extraction d'entités	! JCL !
! DSUSDEXH	! Extraction tables pour habillage DSMS	! JCL !
! DSUSDPRT	! Edition des requêtes	! JCL !
! DSUSDUPT	! Mise à jour de la base DSMS	! JCL !
! DSUSDXBJ	! Extraction du journal DSMS	! JCL !
! DSUSDPDF	! Pre-processeur DAF programmes générés	! JCL !



PROCEDURES D'IMPLANTATION  
 -----

! Membre	! Contenu	!Nature!
! DSINALxx	! Allocation des fichiers (*)	! JCL !
! DSINALIV	! Alloc. librairie d'appel des procédures	! JCL !
! DSINALLI	! Alloc. des librairies	! JCL !
! DSINBLNK	! Link des programmes batch	! JCL !
! DSINBLSO	! Linker modèle pour les programmes batch	! JCL !
! DSINB1LN	! Link des programmes de 'DPRT'	! JCL !
! DSINB2LN	! Link des programmes de 'DEXT'	! JCL !
! DSINB3LN	! Link des programmes de 'DUPT'	! JCL !
! DSINB4LN	! Link des programmes de 'DEXH'	! JCL !
! DSINB5LN	! Link des programmes de 'DXBJ'	! JCL !
! DSINB6LN	! Link des programmes de 'DREN'	! JCL !
! DSINB7LN	! Link du pré-processeur DAF	! JCL !
! DSINDRS1	! Restauration de la base de test	! JCL !
! DSININJB	! Initialisation fichier PACDJB	! JCL !
! DSINMAXI	! Enregistrement maxi pour index	! DAT !
! DSINMGEN	! Lancement procédure génération du TDS	! JCL !
! DSINMPRE	! Lancement procédure préparation du TDS	! JCL !
! DSINDR80	! Reprise 8.0.1	! JCL !
! DSINDR8Q	! Reprise 8.0.2 01/02	! JCL !
! DSINDR8X	! Reprise 8.0.2 compatible PB801	! JCL !
! DSINDR15	! Reprise 1.2 ou 1.5	! JCL !
! DSINDR5J	! Reprise du journal 1.2 ou 1.5	! JCL !
! DSINPRBS	! Re-allocation des fichiers	! JCL !
! DSINPRDS	! Allocation fichiers et bibliothèques	! JCL !
! DSINTLNK	! Link des TPRs dans la SMLIB	! JCL !
! DSINTLSO	! Linker modèle pour les TPRs	! JCL !
! DSINUNLD	! Déchargement bande d'implantation	! JCL !
! DSZZEDIT	! Paramétrage du JCL	! JCL !
! DSZZEXEC	! Paramétrage du JCL	! JCL !
! DSZZJCL	! Paramétrage du JCL	! JCL !
! DSZZVALS	! Paramètres par défaut	! DAT !

(\*) Les deux derniers caractères symbolisés par XX prennent la valeur du nom du fichier. Par exemple, dans le cas de l'allocation du fichier des libellés d'erreurs DE, le JCL à exécuter sera DSINALDE.

APPEL DES PROCEDURES  
 -----

Ces membres contiennent les entrées utilisateur, et sont  
 contenus dans la librairie '\$NMLI.\$LIBINV'.  
 Ils appellent par INVOKE les procédures de la librairie  
 '\$NMLI.\$LIBJCL'.

! Membre	! Contenu	!Nature!
! DSIVDARC	! Archivage du journal	! JCL !
! DSIVDEXP	! Extraction journal	! JCL !
! DSIVDEXQ	! Extraction journal Va Pac < 2.0	! JCL !
! DSIVDEXT	! Extraction d'entités	! JCL !
! DSIVDEXH	! Extraction tables pour habillage DSMS	! JCL !
! DSIVDINI	! Initialisation de la base	! JCL !
! DSIVDLVB	! Remplacement low-values/blancs	! JCL !
! DSIVDPRT	! Edition des requêtes	! JCL !
! DSIVDREO	! Réorganisation de la base	! JCL !
! DSIVDRST	! Restauration de la base	! JCL !
! DSIVDRS1	! Restauration DSMS test	! JCL !
! DSIVDR80	! Reprise 8.0.1	! JCL !
! DSIVDR8Q	! Reprise 8.0.2 01/02	! JCL !
! DSIVDR8X	! Reprise 8.0.2 compatible PB801	! JCL !
! DSIVDR15	! Reprise 1.2 ou 1.5	! JCL !
! DSIVDR5J	! Reprise du journal 1.2 ou 1.5	! JCL !
! DSIVDSAV	! Sauvegarde de la base	! JCL !
! DSIVDUPT	! Mise à jour batch	! JCL !
! DSIVDXBJ	! Extraction du journal	! JCL !
! DSIVDREN	! Changement de codes d'entités	! JCL !
! DSIVDPDF	! Pre-processeur DAF programmes générés	! JCL !
! DSUSDEXP	! Extraction du journal VA Pac	! JCL !
! DSUSDEXQ	! Extr. du journal VA Pac anterieur à 2.0	! JCL !
! DSUSDEXT	! Extraction d'entités	! JCL !
! DSUSDEXH	! Extraction tables pour habillage DSMS	! JCL !
! DSUSDPRT	! Edition des requêtes	! JCL !
! DSUSDUPT	! Mise à jour de la base DSMS	! JCL !
! DSUSDXBJ	! Extraction du journal DSMS	! JCL !

```
" *****  
" * *  
" * DSZZEDIT : CETTE PROCEDURE EDIT EST APPELEE *  
" * PAR L'INTERPRETEUR JCL (DSZZEXEC). *  
" * ELLE PREPARE LE SOUS-FICHER UTILISATEUR *  
" * DSZZVALS POUR L'INTERPRETATION DU JCL. *  
" * *  
" *****  
" DIMINUER LA GRAVITE DES CODES RETOUR  
YW  
" CHARGER DSZZVALS  
RDSZZVALS  
" ENLEVER TOUS LES CARACTERES A BLANC  
GS/ //  
" DETRUIRE LES LIGNES QUI NE COMMENCENT PAS PAR $  
VD/^°C$/  
" INSERER "GS=°C" AU DEBUT DE CHAQUE LIGNE  
GS/^/GS=°C°°CC/  
" INSERER "°C" DEVANT CHAQUE CARACTERE "&"  
GS/°C&/°C°°CC°C&/  
" INSERER "=" A LA FIN DE CHAQUE LIGNE  
GS/$/=/
```

```
COMM 'DSMS 2.5';  
COMM *****;  
COMM * *;  
COMM * DSZZEXEC : PROC. D"INTERPRETATION DU JCL.*;  
COMM * CETTE PROCEDURE PREPARE LE SOUS-FICHIER *;  
COMM * DSZZVALS AVEC LA PROCEDURE DSZZEDIT. *;  
COMM * ELLE REMPLACE ALORS LES PARAMETRES *;  
COMM '* PACBASE PAR LES VALEURS UTILISATEUR DANS *';  
COMM * LES SOUS-FICHIERS CIBLES PAR LE *;  
COMM * PARAMETRE 1, SUIVANT LES CONVENTIONS DE *;  
COMM * CODAGE DES SOUS-FICHIERS DE BIBLIOTHEQUES*;  
COMM * EX : EXEC DSZZEXEC VL=DSZZJCL BRIEF *;  
COMM * *;  
COMM *****;  
ED;  
YB  
B1  
RDSZZEDIT  
B0  
°E1  
Z(JCL)DSZZTEMP  
Q  
STATUS RESET;  
ED LIB:&1;  
YB  
B1  
RDSZZTEMP  
B0  
R &0  
°E1  
Z &0  
Q  
STATUS RESET;  
DELETE DSZZTEMP;
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*';
COMM '* DSZZJCL : CETTE PROCEDURE DOIT ETRE EXECUTEE *';
COMM '* POUR L"INTERPRETATION DU JCL DSMS. *';
COMM '* AVANT L"EXECUTION, DSZZJCL DOIT LUI-MEME *';
COMM '* ETRE INTERPRETE PAR LA PROCEDURE DSZZEXEC. *';
COMM '* EX : EXEC DSZZEXEC VL=DSZZJCL BRIEF; *';
COMM '*';
COMM '*****';
MVL ^DSZZ*,
    CTTUN=' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU' ,
    RFTU=&CTTU$CTTU ,
    CTBSN=' FILESTAT=UNCAT ,DVC=$DVBS ,MD=$MDBS' ,
    RFBS=&CTBS$CTBS ,
    CTLIN=' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI' ,
    RFLI=&CTLI$CTLI ,
    CTBUN=' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU' ,
    RFBU=&CTBU$CTBU ,
    CTDJN=' FILESTAT=UNCAT ,DVC=$DVDJ ,MD=$MDDJ' ,
    RFDJ=&CTDJ$CTDJ ,
    RFTM='DVC=$DVTM ,MD=$MDTM' ;
LMN SL LIB=( $NMLI . $LIBJCL , &RFLI ) ,
    COMFILE=*DS80A ,PRTFILE=DUMMY;
$IN DS80A PRINT JVL;
EXEC DSZZEXEC VL=&1;
$EIN DS80A;
LMN SL LIB=( $NMLI . $LIBJCL , &RFLI ) ,
    COMFILE=*DS80B;
$IN DS80B PRINT JVL;
PR LIB:DSZZVALS;
PR LIB:&1;
$EIN DS80B;
```

```
COMM 'DSMS 2.5';
*****
*      PARAMETRES D'INSTALLATION      *
*      *                               *
*  REMPLACEZ, SI BESOIN EST, LA VALEUR *
*  PAR DEFAUT DE CHAQUE PARAMETRE.     *
*  LES LIGNES PARAMETRE SONT DE LA FORME: *
*      *                               *
*      $NNNNN = VALEUR                  *
*      *                               *
*  CE FICHIER EST TRAITÉ PAR UN         *
*  PROGRAMME EDITEUR QUI CONVERTIT     *
*  LES LIGNES PARAMETRES EN ORDRES     *
*  DE SUBSTITUTIONS.                   *
*      *                               *
*  DONC,                                *
*      *                               *
*  - TOUTES LES LIGNES DONT LE PREMIER *
*  CARACTERE NON BLANC N'EST PAS UN   *
*  DOLLAR ($) SONT TRAITÉES COMME     *
*  COMMENTAIRES.                       *
*  - LE SIGNE EGAL (=) (DELIMITEUR) EST *
*  INTERDIT DANS LA VALEUR DES        *
*  PARAMETRES.                         *
*      *                               *
*****

***** UTILISATEUR D S M S PAR DEFAUT *
$USER = IBM
***** NOM DU CATALOGUE D'IMPLANTATION *
$CTNM = PV
***** NOM DU TDS      D S M S      *
$NMTD = TDS
***** BANDE D'INSTALLATION *****
$TAPE = XXXXXX
$DVTP = CT/M5
***** LANGUE E(NGLISH)/F(RANÇAIS) *****
$LANG = F
***** SUFFIXE BIBLIOTHEQUES *****
***** CU BATCH
$LIBCUB = CUBLIB
***** CU TP
$LIBCUT = CUTLIB
***** CU PERMANENTS
$LIBCUP = CUPLIB
***** JCL
$LIBJCL = JCLLIB
***** LANCEMENT JCLS INVOKE
$LIBINV = INVLIB
***** PRINT DES JCLS APPELES PAR INVOKE **
** BLANC POUR NE PAS LISTER OU &LIST POUR LISTER
$LIST = &LIST
***** LM
$LIBLM = LMLIB
***** SM (VOIR NOTE)
$LIBSM = SMLIB
***** SL TDS (VOIR NOTE)
$LIBSL = SLLIB
***** SL UTILISATEURS
$LIBSU = SULIB
***** REFERENCES FICHIERS BASE *****
***** RADICAL DES FICHIERS DSMS      *
$ROOT = DS
***** IDENTIFIANT DES FICHIERS DSMS  *
```

```
$FILE = 250
***** CISIZE DES FICHIERS BATCH *
** VALEUR PAR DEFAUT POUR UN TYPE DE DISQUE MS/D500 **
$CISEQ = 14336
***** REFERENCES *****
*NMXX = PREFIXE DES FICHIERS
*DVXX = TYPE DE DISQUE
*MDXX = NOM DU DISQUE
*CTXX = CATALOGAGE (Y OU N)
***** FICHIERS UTILISATEURS TDS
$NMTU = DSTU
$DVTU = MS/FSA
$MDTU = DISC01
$CTTU = Y
***** FICHIERS TDS
$DVTD = MS/FSA
$MDTD = DISC02
$CTTD = Y
***** FICHIERS UTILISATEURS BATCH
$NMBU = DSBU
$DVBU = MS/FSA
$MDBU = DISC03
$CTBU = Y
***** FICHIERS BATCH SYSTEME
$NMBS = DSBS
$DVBS = MS/FSA
$MDBS = DISC04
$CTBS = Y
***** FICHIER JOURNAL DSMS
$NMDJ = DSDJ
$DVDJ = MS/FSA
$MDDJ = DISC05
$CTDJ = Y
***** FICHIER TEMPORAIRE
$DVTM = MS/FSA
$MDTM = DISC06
***** BIBLIOTHEQUES
$NMLI = DSLI
$DVLI = MS/FSA
$MDLI = DISC07
$CTLI = Y
***** ENVIRONNEMENT SM (VOIR NOTE) *
*** SI INSERTION DANS UN TDS EXISTANT *
*** PRENDRE DES TPRS NON ATTRIBUEES *
***** NOM DE LA SM DSMS *
$TPR0 = TPR
***** NOM DE LA SM DSMS *
$TPR1 = TPR1
***** RELEASE SYSTEME GCOS7 *
* V6 POUR RELEASE GCOS7 V6 *
* V5 POUR RELEASES ANTERIEURES *
$GCOS7 = V6
*****
* NOTE IMPORTANTE :
*****
* PAR SOUCI D'INTEGRITE AVEC LES *
* PROCEDURES LIVREES PAR LE CONSTRUCTEUR: *
* MTPREP (TP7PREP), MTGEN (TP7GEN), ETC.. *
* IBM PROPOSE UN CHOIX DE VALEURS *
* RESPECTANT LA COHERENCE DU PARAMETRAGE *
* DE CES PROCEDURES. *
* IL EST DONC FORTEMENT RECOMMANDE DE LES *
* APPLIQUER. *
*****
```

## 18.4. DEROULEMENT DE L'IMPLANTATION

### DEROULEMENT DE L'IMPLANTATION

Une fois les JCL obtenus, l'implantation du système DSMS se déroule en plusieurs grandes phases :

- 1 . Allocation des fichiers du système,
- 2 . Préparation du TDS gérant DSMS,
- 3 . Déchargement des fichiers et des programmes,
- 4 . Génération du TDS gérant DSMS,
- 5 . Link-edit des programmes batch et link-edit des TPRs dans la SMLIB prévue pour DSMS,
- 6 . Création d'une base DSMS de tests,
- 7 . Lancement du TDS.



1. ALLOCATION DES FICHIERS DU SYSTEME (DSINPRDS)

L'allocation des fichiers du système s'effectue par l'exécution du membre 'DSINPRDS' contenu dans \$NMLI.\$LIBJCL. Bien que l'allocation de certains fichiers se retrouve dans des procédures exécutées ci-après, cette phase permet de vérifier la bonne disponibilité de la place disque. Ce membre est en fait une suite de PREALLOC et de LIBALLOC, et peut logiquement être découpé de la manière suivante :

. Allocation des fichiers permanents UFAS :

\$NMBS.\$ROOT\$ROOTDE0

. Allocation des fichiers UFAS de la base de tests :

\$NMBU.\$ROOT\$FILEBB  
\$NMBU.\$ROOT\$FILEBJ  
\$NMBU.\$ROOT\$FILEBQ  
\$NMTU.\$ROOT\$FILEDA  
\$NMTU.\$ROOT\$FILEDC  
\$NMDJ.\$ROOT\$FILEDJ  
\$NMTU.\$ROOT\$FILEDX  
\$NMTU.\$ROOT\$ROOTDE

. Allocation des librairies :

\$NMLI.\$LIBLM  
\$NMLI.\$LIBCUB  
\$NMLI.\$LIBCUT  
\$NMLI.\$LIBCUP  
\$NMLI.\$LIBSU  
\$NMLI.\$LIBINV  
\$NMLI.SOURCE  
\$NMLI.ABOUT

LE SOURCE DE LA PROCEDURE

```

COMM 'DSMS 2.5';
COMM '*****';
COMM '* ALLOCATION DES FICHIERS *';
COMM '* ET BIBLIOTHEQUES POUR *';
COMM '* DSMS *';
COMM '*****';
MVL CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    RFTU=&CTTU$CTTU,
    CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
    RFBS=&CTBS$CTBS,
    CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
    RFBU=&CTBU$CTBU,
    CTDJN=' FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
    RFDJ=&CTDJ$CTDJ,
    RFTM=' DVC=$DVTM,MD=$MDTM';
OVL HOLD;
COMM '*** BATCH SYSTEM ***';
IV DSINALD0,($NMLI.$LIBJCL,&RFLI);
COMM '*** TDS ***';
IV DSINALDA,($NMLI.$LIBJCL,&RFLI);
IV DSINALDC,($NMLI.$LIBJCL,&RFLI);
IV DSINALDE,($NMLI.$LIBJCL,&RFLI);
IV DSINALDJ,($NMLI.$LIBJCL,&RFLI);
IV DSINALDX,($NMLI.$LIBJCL,&RFLI);
COMM '*** BATCH ***';
IV DSINALBB,($NMLI.$LIBJCL,&RFLI);
IV DSINALBJ,($NMLI.$LIBJCL,&RFLI);
IV DSINALBQ,($NMLI.$LIBJCL,&RFLI);
COMM '*** LIBRARIES ***';
IV DSINALLI,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.$LIBLM,LM,5,1,70);
IV DSINALLI,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.$LIBCUB,CU,3,1,70);
IV DSINALLI,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.$LIBCUT,CU,6,1,100);
IV DSINALLI,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.$LIBCUP,CU,1,1,10);
IV DSINALLI,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.$LIBSU,SL,1,1,100);
IV DSINALIV,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.$LIBINV,SL,1,1,50);
IV DSINALLI,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.SOURCE,SL,1,1,50);
IV DSINALLI,($NMLI.$LIBJCL,&RFLI) VL=($NMLI.ABOUT,SL,1,1,50);

```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALLI          *';
COMM '*          =====          *';
COMM '*  ALLOCATION DES LIBRAIRIES DSMS *';
COMM '*****';
MVL   CTLI=' FILESTAT=CAT,UNCATNOW',
      CLIY=' FILESTAT=CAT,DVC=$DVLI,MD=$MDLI',
      DV='DVC=$DVLI',MD='MD=$MDLI',
      COMP&2=COMPACT;
OVL   HOLD;
JUMP  CAT$CTLI;
CATY:
DALC  &1,&CTLI;
JUMP  CONTINUE;
CAT   &1,TYPE=FILE,SHARE=UNSPEC;
LALC  &2,(&1,&CLiy,SZ=(&3,&4)),MEMBERS=&5,&COMPSL;
JUMP  END;
CATN:
DALC  &1,&DV,&MD;
JUMP  CONTINUE;
LALC  &2,(&1,&DV,&MD,SZ=(&3,&4)),MEMBERS=&5,&COMPSL;
END:
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALIV          *';
COMM '*          =====          *';
COMM '*  ALLOCATION LIBRAIRIE INVOKE PROC DSMS *';
COMM '*****';
MVL  CTLI=' FILESTAT=CAT,UNCATNOW',
     CLIY=' FILESTAT=CAT,DVC=$DVLI,MD=$MDLI',
     CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     COMP&2=COMPACT;
OVL  HOLD;
JUMP CAT$CTLI;
CATY:
CAT  &1,TYPE=FILE,SHARE=UNSPEC;
JUMP CONTINUE;
LALC &2,(&1,&CLiy,SZ=(&3,&4)),MEMBERS=&5,&COMPSL;
JUMP CONTINUE;
JUMP CATEND;
CATN:
LALC &2,(&1,&CTLIN,SZ=(&3,&4)),MEMBERS=&5,&COMPSL;
JUMP CONTINUE;
CATEND:
LIB  SL,INLIB1=($NMLI.$LIBJCL,&RFLI);
LMN  SL,LIB=(&1,&RFLI),
     COM='MV IL1:DSIV*,REPLACE;';
END:
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALD0          *';
COMM '*          =====          *';
COMM '*  ALLOCATION DU FICHIER DE0  *';
COMM '*****';
MVL  CTBSN= ' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS ',
      CTBSY= ' FILESTAT=CAT,UNCATNOW ',RFBS=&CTBS$CTBS,
      CBSN= ' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS ',
      CBSY= ' FILESTAT=CAT ',RBS=&CBS$CTBS,
      CTLIN= ' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI ',
      RFLI=&CTLI$CTLI,
      DV1= ' DVC=$DVBS ',MD1= ' MD=$MDBS ',
      CATFN= ' FILESTAT=UNCAT ',
      CATFY= ' FILESTAT=CAT ';
DALC $NMBS.$ROOT$ROOTDE0,&RFBS;
JUMP CONTINUE;
JUMP D0$CTBS;
DOY:
CAT  $NMBS.$ROOT$ROOTDE0,TYPE=FILE,SHARE=UNSPEC;
DON:
PALC $NMBS.$ROOT$ROOTDE0,
      UNIT=CYL,&DV1,GBL=( &MD1,SZ=8 ),
      UFAS=( SEQ=( CISZ=2048,RECSZ=90 ) ),
      &CATF$CTBS;
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALBB                      *';
COMM '*          =====                      *';
COMM '*  ALLOCATION DU FICHER BB                *';
COMM '*****';
MVL  CTBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
      CTBUY= ' FILESTAT=CAT ,UNCATNOW ' ,RFBU=&CTBU$CTBU ,
      CBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
      CBUY= ' FILESTAT=CAT ' ,RBU=&CBU$CTBU ,
      CTLIN= ' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      DV1= ' DVC=$DVBU ' ,MD1= ' MD=$MDBU ' ,
      CATFN= ' FILESTAT=UNCAT ' ,
      CATFY= ' FILESTAT=CAT ' ;
DALC $NMBU.$ROOT$FILEBB ,&RFBU;
JUMP CONTINUE;
DALC $NMBU.$ROOT$FILEBB/G-1 ,&RFBU;
JUMP CONTINUE;
UNCAT $NMBU.$ROOT$FILEBB ,TYPE=FILE;
JUMP CONTINUE;
JUMP BB$CTBU;
BBY:
CAT  $NMBU.$ROOT$FILEBB ,TYPE=FILE ,NBGEN=2 ,SHARE=UNSPEC;
BBN:
PALC $NMBU.$ROOT$FILEBB/G+1 ,
      UNIT=CYL ,&DV1 ,GBL= ( &MD1 ,SZ=1 ) ,INCRSZ=1 ,
      UFAS= ( SEQ= ( CISZ=2048 ,RECSZ=350 ,RECFORM=V ) ) ,
      &CATF$CTBU;
PALC $NMBU.$ROOT$FILEBB/G+1 ,
      UNIT=CYL ,&DV1 ,GBL= ( &MD1 ,SZ=1 ) ,INCRSZ=1 ,
      UFAS= ( SEQ= ( CISZ=2048 ,RECSZ=350 ,RECFORM=V ) ) ,
      &CATF$CTBU;
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALBJ                      *';
COMM '*          =====                      *';
COMM '*    ALLOCATION DU FICHIER BJ              *';
COMM '*****';
MVL  CTBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
      CTBUY= ' FILESTAT=CAT ,UNCATNOW ' ,RFBU=&CTBU$CTBU ,
      CBUN= ' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU ' ,
      CBUY= ' FILESTAT=CAT ' ,RBU=&CBU$CTBU ,
      CTLIN= ' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      DV1= ' DVC=$DVBU ' ,MD1= ' MD=$MDBU ' ,
      CATFN= ' FILESTAT=UNCAT ' ,
      CATFY= ' FILESTAT=CAT ' ;
DALC $NMBU.$ROOT$FILEBJ ,&RFBU;
JUMP CONTINUE;
DALC $NMBU.$ROOT$FILEBJ/G-1 ,&RFBU;
JUMP CONTINUE;
UNCAT $NMBU.$ROOT$FILEBJ ,TYPE=FILE;
JUMP CONTINUE;
JUMP BJ$CTBU;
BJY:
CAT  $NMBU.$ROOT$FILEBJ ,TYPE=FILE ,NBGEN=2 ,SHARE=UNSPEC;
BJN:
PALC $NMBU.$ROOT$FILEBJ/G+1 ,
      UNIT=CYL ,&DV1 ,GBL= ( &MD1 ,SZ=1 ) ,INCRSZ=1 ,
      UFAS= ( SEQ= ( CISZ=2048 ,RECSZ=180 ,RECFORM=F ) ) ,
      &CATF$CTBU;
PALC $NMBU.$ROOT$FILEBJ/G+1 ,
      UNIT=CYL ,&DV1 ,GBL= ( &MD1 ,SZ=1 ) ,INCRSZ=1 ,
      UFAS= ( SEQ= ( CISZ=2048 ,RECSZ=180 ,RECFORM=F ) ) ,
      &CATF$CTBU;
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALBQ          *';
COMM '*          =====          *';
COMM '*  ALLOCATION DU FICHIER BQ    *';
COMM '*****';
MVL  CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      CTBUY=' FILESTAT=CAT,UNCATNOW',RFBU=&CTBU$CTBU,
      CBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      CBUY=' FILESTAT=CAT',RBU=&CBU$CTBU,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      DV1='DVC=$DVBU',MD1='MD=$MDBU',
      CATFN=' FILESTAT=UNCAT',
      CATFY=' FILESTAT=CAT';
DALC $NMBU.$ROOT$FILEBQ,&RFBU;
JUMP CONTINUE;
JUMP BQ$CTBU;
BQY:
CAT  $NMBU.$ROOT$FILEBQ,TYPE=FILE,SHARE=UNSPEC;
BQN:
PALC $NMBU.$ROOT$FILEBQ,
      UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
      UFAS=(SEQ=(CISZ=2048,RECSZ=180,RECFORM=F)),
      &CATF$CTBU;
```



```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALDA          *';
COMM '*          =====          *';
COMM '*  ALLOCATION DU FICHIER DA    *';
COMM '*****';
MVL  CTTUN= ' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU ' ,
      CTTUY= ' FILESTAT=CAT ,UNCATNOW ' ,RFTU=&CTTU$CTTU ,
      CTUN= ' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU ' ,
      CTUY= ' FILESTAT=CAT ' ,RTU=&CTTU$CTTU ,
      CTLIN= ' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI ' ,
      RFLI=&CTLI$CTLI ,
      DV1= ' DVC=$DVTU ' ,MD1= ' MD=$MDTU ' ,
      CATFN= ' FILESTAT=UNCAT ' ,
      CATFY= ' FILESTAT=CAT ' ;
DALC $NMTU.$ROOT$FILEDA ,&RFTU;
JUMP CONTINUE;
JUMP DA$CTTU;
DAY:
CAT  $NMTU.$ROOT$FILEDA ,TYPE=FILE ,SHARE=UNSPEC;
DAN:
PALC $NMTU.$ROOT$FILEDA ,
      UNIT=CYL ,&DV1 ,GBL=( &MD1 ,SZ=1 ) ,INCRSZ=1 ,
      UFAS=( INDEXED=( CISZ=2048 ,RECSZ=340 ,RECFORM=V ,
                      KEYLOC=3 ,KEYSZ=40 ) ) ,
      &CATF$CTTU;
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALDC                      *';
COMM '*          =====                      *';
COMM '*  ALLOCATION DU FICHIER DC                *';
COMM '*****';
MVL  CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      CTTUY=' FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
      CTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      CTUY=' FILESTAT=CAT',RTU=&CTU$CTTU,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      DV1='DVC=$DVTU',MD1='MD=$MDTU',
      CATFN=' FILESTAT=UNCAT',
      CATFY=' FILESTAT=CAT';
DALC $NMTU.$ROOT$FILEDC,&RFTU;
JUMP CONTINUE;
JUMP DC$CTTU;
DCY:
CAT  $NMTU.$ROOT$FILEDC,TYPE=FILE,SHARE=UNSPEC;
DCN:
PALC $NMTU.$ROOT$FILEDC,
      UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
      UFAS=(INDEXED=(CISZ=4096,RECSZ=168,RECFORM=V,
                     KEYLOC=3,KEYSZ=31)),
      &CATF$CTTU;
CR   IF=( $NMLI.$LIBJCL,&RFLI,SUBFILE=DSINMAXI),
      OF=( $NMTU.$ROOT$FILEDC,&RFTU),
      COMFILE=( $NMLI.$LIBJCL,&RFLI,SUBFILE=DSEXPDSL),START=2;
END:
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALDX          *';
COMM '*          =====          *';
COMM '*  ALLOCATION DU FICHIER DX    *';
COMM '*****';
MVL  CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      CTTUY=' FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
      CTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      CTUY=' FILESTAT=CAT',RTU=&CTTU$CTTU,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      DV1='DVC=$DVTU',MD1='MD=$MDTU',
      CATFN=' FILESTAT=UNCAT',
      CATFY=' FILESTAT=CAT';
DALC $NMTU.$ROOT$FILEDX,&RFTU;
JUMP CONTINUE;
JUMP DX$CTTU;
DXY:
CAT  $NMTU.$ROOT$FILEDX,TYPE=FILE,SHARE=UNSPEC;
DXN:
PALC $NMTU.$ROOT$FILEDX,
      UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
      UFAS=(INDEXED=(CISZ=2048,RECSZ=80,RECFORM=F,
                     KEYLOC=1,KEYSZ=50)),
      &CATF$CTTU;
```

```
COMM 'DSMS 2.5';
MVL CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    CTTUY='FILESTAT=CAT,UNCATNOW',RFTU=&CTTU$CTTU,
    CTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    CTUY='FILESTAT=CAT',RTU=&CTU$CTTU,
    CTTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    DV1='DVC=$DVTU',MD1='MD=$MDTU',
    CATFN='FILESTAT=UNCAT',
    CATFY='FILESTAT=CAT';
DALC $NMTU.$ROOT$ROOTDE,&RFTU;
JUMP CONTINUE;
JUMP DE$CTTU;
DEY:
CAT $NMTU.$ROOT$ROOTDE,TYPE=FILE,SHARE=UNSPEC;
DEN:
PALC $NMTU.$ROOT$ROOTDE,
    UNIT=CYL,&DV1,GBL=(&MD1,SZ=8),INCRSZ=1,
    UFAS=(INDEXED=(CISZ=2048,RECSZ=90,KEYLOC=1,KEYSZ=17)),
    &CATF$CTTU;
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINALDJ          *';
COMM '*          =====          *';
COMM '*****';
MVL  CTDJN=' FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
      CTDJY=' FILESTAT=CAT,UNCATNOW',RFDJ=&CTDJ$CTDJ,
      CDJN=' FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
      CDJY=' FILESTAT=CAT',RDJ=&CDJ$CTDJ,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      DV1='DVC=$DVDJ',MD1='MD=$MDDJ',
      CATFN=' FILESTAT=UNCAT',
      CATFY=' FILESTAT=CAT';
DALC $NMDJ.$ROOT$FILEDJ,&RFDJ;
JUMP CONTINUE;
JUMP DJ$CTDJ;
DJY:
CAT  $NMDJ.$ROOT$FILEDJ,TYPE=FILE,SHARE=UNSPEC;
DJN:
PALC $NMDJ.$ROOT$FILEDJ,
      UNIT=CYL,&DV1,GBL=(&MD1,SZ=1),INCRSZ=1,
      UFAS=(RELATIVE=(CISZ=2048,RECSZ=180)),
      &CATF$CTDJ;
```

2. PREPARATION DU TDS (DSINMPRE)

1er cas : création d'un TDS DSMS indépendant.

La préparation du TDS DSMS s'effectue par exécution de la procédure système TP7PREP fournie par le constructeur. Le membre 'DSINMPRE' de la librairie \$NMLI.\$LIBJCL prévoit l'exécution de la procédure TP7PREP (ou MTPREP pour GCOS7 V5) livrée par le constructeur dans la librairie SYS.HSLLIB.

2ème cas : insertion de DSMS dans un TDS existant.

Il suffit, lors de cette phase, d'initialiser au niveau de la SMLIB associée, les TPRs qui recevront les compil-units des programmes transactionnels.

REMARQUE :

La librairie \$NMTD.\$LIBSL est généralement allouée lors de cette phase préparatoire. Les procédures DSMS d'installation ne prévoient donc pas son allocation. Quelle que soit la solution adoptée par l'utilisateur, celui-ci doit s'assurer de son existence avant l'exécution de la phase suivante.

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*';
COMM '*          PREPARATION DU TDS          *';
COMM '*';
COMM '* IMPORTANT :';
COMM '* LE PARAMETRE "SYSFILE" INDIQUE L'ETAT DES *';
COMM '* TROIS FICHIERS SYSTEME DU TDS $NMTD *';
COMM '* ($NMTD.CTLM, $NMTD.CTLN, $NMTD.RECOV) *';
COMM '*';
COMM '* - SYSFILE=CAT POUR OPTION CATALOGAGE *';
COMM '* - SYSFILE=RSD POUR OPTION RESIDENT *';
COMM '* - SYSFILE=RSDN POUR OPTION NON RESIDENT *';
COMM '*****';
OVL HOLD;
VL PRY='SYSFILE=CAT,FILESTAT=CAT,CATNAME=$CTNM,IMPORT=NO',
PREV6=TP7PREP,
PRN='SYSFILE=RSD,FILESTAT=UNCAT',
FF='$NMTD,$DVTD,$MDTD,$DVTD,$MDTD,DEAL=Y',
GG='DBGSZ=1,MAXDBG=3,CBLSZ=1,SMSZ=7,MAXSM=10',
VLVL='VL=('&FF','&PR$CTTD)';
IV &PREV6 SYS.HSLLIB &VLVL,&GG);
SEND '===> PREPARATION OF '$NMTD' SUCCESSFUL <===';
```

3. IMPLANTATION DES FICHIERS ET DES PROGRAMMES (DSINUNLD)

L'implantation des fichiers et des programmes s'effectue par l'exécution du membre 'DSINUNLD' de la librairie \$NMLI.\$LIBJCL.

Ce membre est une suite de LIBMAINT et de CREATE. Il peut logiquement être découpé de la manière suivante :

- . Déchargement des sources COBOL :

\$NMTD.\$LIBSL

- . Déchargement des programmes (compil-units) :

\$NMLI.\$LIBCUB

\$NMLI.\$LIBCUT

- . Copie des CU des sous-programmes TP dans la librairie permanente :

\$NMLI.\$LIBCUP

- . Copie du fichier des libellés d'erreurs

\$NMBS.\$ROOT\$ROOTDE0 et \$NMTU.\$ROOT\$ROOTDE

- . Copie de la bibliothèque des sources de contrôle

\$NMLI.SOURCE

- . Copie de la bibliothèque de versions.

\$NMLI.ABOUT



LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*      DECHARGEMENT DE LA BANDE      *';
COMM '*      D"IMPLANTATION                 *';
COMM '*      BIBLIOTHEQUES ET FICHIERS     *';
COMM '*      *                               *';
COMM '*****';
MVL  LANG=F,
     DVT='DVC=$DVTP',MDT='MD=$TAPE',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTTDN='FILESTAT=UNCAT,DVC=$DVTD,MD=$MDTD',
     RFTD=&CTTD$CTTD,
     RFTM='DVC=$DVTM,MD=$MDTM';
LMN  SL,IF=(SV.SL,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=( $NMTD.$LIBSL,&RFTD),
     COM='MV INFILE:*,REPLACE;';
LMN  CU,IF=(SV.CUB,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=( $NMLI.$LIBCUB,&RFLI),
     COM='MV INFILE:*,REPLACE;';
LMN  CU,IF=(SV.CUT,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=( $NMLI.$LIBCUT,&RFLI),
     COM='MV INFILE:*,REPLACE;';
LIB  CU,IL1=( $NMLI.$LIBCUT,&RFLI);
LMN  CU,LIB=( $NMLI.$LIBCUP,&RFLI),
     COM='MV IL1:DSCHOI,REPLACE;STATUS RESET;
          MV IL1:ZAR980,REPLACE;STATUS RESET;
          MV IL1:ZAR985,REPLACE;STATUS RESET;
          MV IL1:PDTPDF,REPLACE;STATUS RESET;';
CR   IF=(SV.DE0,&DVT,&MDT,FSN=ANY,END=LEAVE),
     OF=( $NMBS.$ROOT$ROOTDE0,&RFBS);
CR   IF=( $NMBS.$ROOT$ROOTDE0,&RFBS),
     OF=( $NMTU.$ROOT$ROOTDE,&RFTU);
LMN  SL,IF=(SV.SOURCE,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=( $NMLI.SOURCE,&RFLI),
     COM='MV INFILE:*,REPLACE;';
LMN  SL,IF=(SV$LANG.ABOUT,&DVT,&MDT,FSN=ANY,END=LEAVE),
     LIB=( $NMLI.ABOUT,&RFLI),
     COM='MV INFILE:*,REPLACE;';
```

4. GENERATION DU TDS DSMS (DSINMGEN)

On ne traitera dans ce paragraphe que la génération d'un TDS DSMS indépendant.

Cette génération s'effectue par exécution de la procédure 'DSINMGEN' de la librairie \$NMLI.\$LIBJCL.

ATTENTION :

-----

Avant la génération du TDS l'utilisateur doit contrôler le source TDS livré dans la librairie \$NMTD.\$LIBSL et le renommer ou le recopier sous le nom réservé 'STDS', en changeant la clause "PROGRAM-ID" par \$NMTD (par défaut : DS20)

STDSD : source TDS DSMS

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '* *';
COMM '* GENERATION DU TDS *';
COMM '* *';
COMM '*****';
MVL CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    RFTU=&CTTU$CTTU,
    TDDVN=$DVRTD,TDMDN=$MDTD,
    GENV6=TP7GEN,
    CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
    RFBS=&CTBS$CTBS,
    CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    LMDVN=$DVLI,LMDN=$MDLI,
    CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
    RFBU=&CTBU$CTBU,
    CTDJN=' FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
    RFDJ=&CTDJ$CTDJ,
    CTTDN=' FILESTAT=UNCAT,DVC=$DVRTD,MD=$MDTD',
    RFTD=&CTTD$CTTD,
    RFTM=' DVC=$DVTM,MD=$MDTM';
OVL HOLD;
LIB CU,INLIB1=( $NMLI.$LIBCUP,&RFLI);
LMN CU,LIB=TEMP,
    COM=' MV IL1:DSCHOI;STATUS RESET;
        MV IL1:PDPDF;STATUS RESET;
        MV IL1:ZAR980;STATUS RESET;
        MV IL1:ZAR985;STATUS RESET;';
IV &GENV6 SYS.HSLLIB
    VL=( $NMTD,&TDDV$CTTD,&TDMD$CTTD,,,,,$NMLI.$LIBLM,
        LMDVC=&LMDV$CTLI,LMD=&LMD$CTLI);
```

5. LINK-EDIT DES PROGRAMMES

Le link-edit des programmes passe par l'exécution de 9 membres de la librairie \$NMLI.\$LIBJCL :

- . Link-edit des programmes transactionnels par l'exécution du membre 'DSINTLNK'.
- . Link-edit des programmes de la procédure d'édition des requêtes utilisateurs (DPRT) par l'exécution du membre 'DSINB1LN'.
- . Link-edit des programmes de la procédure d'extraction d'entités (DEXT) par l'exécution du membre 'DSINB2LN'.
- . Link-edit des programmes de la procédure de mise à jour batch (DUPT) par l'exécution du membre 'DSINB3LN'.
- . Link-edit des programmes de la procédure d'extraction des tables pour habillage DSMS par l'exécution du membre 'DSINB4LN'.
- . Link-edit des programmes de la procédure d'extraction du journal archivé DSMS par l'exécution du membre 'DSINB5LN'.
- . Link-edit des programmes de la procédure de changement de codes et mots clés (DREN) : membre 'DSINB6LN'.
- . Link-edit du programme pré-processeur DAF : 'DSINB7LN'
- . Link-edit des autres programmes batch par l'exécution du membre 'DSINBLNK'.

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINTLNK          *';
COMM '*          =====          *';
COMM '* LINK DE TOUS LES PROGRAMMES TP. *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '* $TPR0 : NOM DE LA TPR D S M S DANS SMLIB*';
COMM '* $TPR1 : NOM DE LA TPR D S M S DANS SMLIB*';
COMM '* * * * * * * * * * * * * * * * *';
COMM '* GARDEZ L"EDITION DES LINKS *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '*****';
MVL TPR3=' $TPR0',TPR4=' $TPR1',
    CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    RFTU=&CTTU$CTTU,
    CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
    RFBS=&CTBS$CTBS,
    CTTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
    RFBU=&CTBU$CTBU,
    CTDJN=' FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
    RFDJ=&CTDJ$CTDJ,
    RFTM=' DVC=$DVTM,MD=$MDTM';
OVL HOLD;
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS0AA0,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00AA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00AB,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00BA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00B1,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00B2,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00B3,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00B4,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00B5,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00EA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00E1,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00E2,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00E3,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00E4,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00E5,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00E6,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00FA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00FB,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00HE,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00JO,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00KA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00K1,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00K2,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00K3,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00LE,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00LS,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00MA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00PA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00P1,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00QA,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00QB,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00QC,&TPR3);
IV DSINTLSO,($NMLI.$LIBJCL,&RFLI),VL=(DS00Q1,&TPR3);
```

IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q2,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q3,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q4,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q5,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q6,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q7,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q8,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00Q9,&TPR3);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00SA,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00SI,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S1,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S3,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S4,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S5,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S6,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S7,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S8,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00S9,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TA,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TT,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TU,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TV,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TW,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TX,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TY,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00TZ,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T1,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T2,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T3,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T4,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T5,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T6,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T7,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T8,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00T9,&TPR4);  
IV DSINTLSO, (\$NMLI.\$LIBJCL,&RFLI),VL=(DS00UD,&TPR4);

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINTLSO          *';
COMM '*          =====          *';
COMM '* LINKER MODELE POUR LES PROGRAMMES TP. *';
COMM '* APPELE PAR INVOKE DEPUIS DSINTLNK *';
COMM '*****';
MVL  CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      RFTU=&CTTU$CTTU,
      LINKV6=TP7LINK&2,
      CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      RFBU=&CTBU$CTBU,
      CTTDN=' FILESTAT=UNCAT,DVC=$DVTD,MD=$MDTD',
      RFTD=&CTTD$CTTD,
      RFTM='DVC=$DVTM,MD=$MDTM';
LIB  CU,INLIB1=( $NMLI.$LIBCUT,&RFLI);
LK   &1,SM,OUTLIB=( $NMTD.$LIBSM,&RFTD),
      COMFILE=( $NMTD.$LIBSL,&RFTD,SUBFILE=&LINKV6);
JUMP CONT,SEV,LE,1;JUMP CONTINUE;
SEND 'DSINTLNK: '&1' NOT LINKED INTO '&2'.';
CONT:
END:
```

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINB1LN          *';
COMM '*          =====          *';
COMM '* LINK DES PROGRAMMES DE LA PROCEDURE *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '* D EDITION DES REQUETES (DPRT) *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '*****';
MVL  CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     PGE='E',PGLNG=&PG$LANG;
OVL  HOLD;
LIB  CU,INLIB1=( $NMLI.$LIBCUB,&RFLI);
LK   PDSB&PGLNG,OUTLIB=( $NMLI.$LIBLM,&RFLI),
     COM=' INCLUDE=( PDSA10,PDSRQ0,PDSRQ1,PDSRQ2,
     PDSRQ3,PDSE90,PDSFAC,PDSUAA)';
END:
```



LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINB2LN          *';
COMM '*          =====          *';
COMM '* LINK DES PROGRAMMES DE LA PROCEDURE *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '* D EXTRACTION D ENTITES (DEXT) *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '*****';
MVL  CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     PGE='E',PGLNG=&PG$LANG;
OVL  HOLD;
LIB  CU,INLIB1=( $NMLI.$LIBCUB,&RFLI);
LK   PDSEX&PGLNG,OUTLIB=( $NMLI.$LIBLM,&RFLI),
     COM=' INCLUDE=( PDSXCT,PDSRQ0,PDSRQ1,PDSXST,
     PDSXTR,PDSFAC,PDSUAA);';
END:
```

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINB3LN          *';
COMM '*          =====          *';
COMM '* LINK DES PROGRAMMES DE LA PROCEDURE *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '* DE MISE A JOUR BATCH (DUPT) *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '*****';
MVL CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI ',
RFLI=&CTLI$CTLI;
OVL HOLD;
LIB CU,INLIB1=( $NMLI.$LIBCUB,&RFLI );
LK PDSUP0,OUTLIB=( $NMLI.$LIBLM,&RFLI ),
COM=' INCLUDE=( PDCHOI,PDSFAC,PDSUAA,PDSUB1,
PDSUB2,PDSUB3,PDSUB4,PDSUE1,PDSUE2,PDSUE3,
PDSUK1,PDSUQ1,PDSUQ2,PDSUQ5,PDSUQ6,PDSUQ7,
PDSUQ8,PDSUS1,PDSUS3,PDSUS4,PDSUS6,PDSUSI,PDSUT1,
PDSUT2,PDSUT3,PDSUT4,PDSUT5,PDSUT6,PDSUT7,
PDSUT8,PDSUT9,PDSUTT,PDSUTV,PDSUTW,PDSUTX,
PDSUTY,PDSUTZ,PDSUP1)';
END:
```

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINB4LN          *';
COMM '*          =====          *';
COMM '* LINK DES PROGRAMMES DE LA PROCEDURE *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '* EXTRACTION DES TABLES (DEXH) *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '*****';
MVL CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI ',
RFLI=&CTLI$CTLI;
OVL HOLD;
LIB CU,INLIB1=( $NMLI.$LIBCUB,&RFLI );
LK PDSXTH,OUTLIB=( $NMLI.$LIBLM,&RFLI ),
COM=' INCLUDE=( PDSFAC,PDSUAA );';
END;
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINB5LN          *';
COMM '*          =====          *';
COMM '*  LINK DES PROGRAMMES DE LA PROCEDURE *';
COMM '*                                     *';
COMM '*  EXTRACTION DU JOURNAL  (DXBJ)      *';
COMM '*                                     *';
COMM '*****';
MVL  CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI;
OVL  HOLD;
LIB  CU,INLIB1=( $NMLI.$LIBCUB,&RFLI);
LK   PDS700,OUTLIB=( $NMLI.$LIBLM,&RFLI),
     COM=' INCLUDE=( PDSFAC,PDSUAA)';
END:
```

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINB6LN          *';
COMM '*          =====          *';
COMM '*          *';
COMM '* LINK DES PROGRAMMES DE LA PROCEDURE DREN *';
COMM '*          *';
COMM '*****';
MVL  CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     PGE='E',PGLNG=&PG$LANG;
OVL  HOLD;
LIB  CU,INLIB1=( $NMLI.$LIBCUB,&RFLI);
LK   PDSMS&PGLNG,OUTLIB=( $NMLI.$LIBLM,&RFLI),
     COM='INCLUDE=( PDSRCT,PDSRMS,PDSRFU,PDSJMS,PDSFAC,PDSUAA)';
END:
```

```
COMM 'DSMS 2.5';  
COMM '*****';  
COMM '*          DSINB7LN          *';  
COMM '*          =====          *';  
COMM '*****';  
MVL  CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',  
      RFLI=&CTLI$CTLI;  
OVL  HOLD;  
LIB  CU,INLIB1=($NMLI.$LIBCUB,&RFLI);  
LK   DAFD10,OUTLIB=($NMLI.$LIBLM,&RFLI),  
      COM='INCLUDE=PDSFAC;';  
END:
```

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINBLNK          *';
COMM '*          =====          *';
COMM '* LINK DE TOUS LES PROGRAMMES BATCH *';
COMM '* * * * * * * * * * * * * * * * *';
COMM '* GARDEZ L"EDITION DES LINKS *';
COMM '* * * * * * * * * * * * * * * * *';
MVL  CTTUN=' FILESTAT=UNCAT, DVC=$DVTU, MD=$MDTU ',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT, DVC=$DVBS, MD=$MDBS ',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT, DVC=$DVLI, MD=$MDLI ',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT, DVC=$DVBU, MD=$MDBU ',
      RFBU=&CTBU$CTBU,
      CTDJN=' FILESTAT=UNCAT, DVC=$DVDJ, MD=$MDDJ ',
      RFDJ=&CTDJ$CTDJ,
      RFTM=' DVC=$DVTM, MD=$MDTM ';
OVL  HOLD;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSBAS;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSINI;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSLVB;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR10;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR20;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR30;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR40;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR8B;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR8C;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR8X;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR8Q;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR8R;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR15;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDSR5J;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=REP2PJ;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS300;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS320;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS380;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS381;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS400;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS450;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS500;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS600;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS610;
IV   DSINBLSO, ($NMLI.$LIBJCL,&RFLI), VL=PDS900;
```

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*          DSINBLSO          *';
COMM '*          =====          *';
COMM '* LINKER MODELE POUR LES PROGRAMMES *';
COMM '* BATCH APPELE PAR INVOKE DEPUIS *';
COMM '* PBINBLNK. *';
COMM '*****';
MVL CTTUN=' FILESTAT=UNCAT ,DVC=$DVTU ,MD=$MDTU' ,
    RFTU=&CTTU$CTTU ,
    CTBSN=' FILESTAT=UNCAT ,DVC=$DVBS ,MD=$MDBS' ,
    RFBS=&CTBS$CTBS ,
    CTTLIN=' FILESTAT=UNCAT ,DVC=$DVLI ,MD=$MDLI' ,
    RFLI=&CTLI$CTLI ,
    CTBUN=' FILESTAT=UNCAT ,DVC=$DVBU ,MD=$MDBU' ,
    RFBU=&CTBU$CTBU ,
    CTDJN=' FILESTAT=UNCAT ,DVC=$DVDJ ,MD=$MDDJ' ,
    RFDJ=&CTDJ$CTDJ ,
    RFTM=' DVC=$DVTM ,MD=$MDTM' ;
LIB CU ,INLIB1=( $NMLI . $LIBCUB ,&RFLI ) ;
LK &1 ,OUTLIB=( $NMLI . $LIBLM ,&RFLI ) ;
JUMP CONT ,SEV ,LE ,1 ; JUMP CONTINUE ;
SEND 'DSINBLNK: '&1' NOT LINKED' ;
CONT :
END :
```



6. CREATION D'UNE BASE DSMS DE TEST (DSINDRS1)

La création de la base DSMS de test s'effectue par exécution du membre DSIVDRS1 (procédure de restauration DSMS) qui utilise en entrée la sauvegarde contenue sur la bande d'implantation (Nom physique: SV.BB).

LE SOURCE DE LA PROCEDURE

```

COMM 'DSMS 2.5';
MVL PACDMB='MBDRST',
    DVT='DVC=$DVTP',MDT='MD=$TAPE',
    CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    RFTU=&CTTU$CTTU,
    CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
    RFBS=&CTBS$CTBS,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
    RFBU=&CTBU$CTBU,
    CTDJN='FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
    RFDJ=&CTDJ$CTDJ,
    RFTM='DVC=$DVTM,MD=$MDTM';
CR IF=*DRS1,
    OF=(TMBREST,TEMPRY,&RFTM,END=PASS),
    OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : DA,DC,DJ,DX ***';
IV DSINALDA ($NMLI.$LIBJCL,&RFLI);
IV DSINALDC ($NMLI.$LIBJCL,&RFLI);
IV DSINALDJ ($NMLI.$LIBJCL,&RFLI);
IV DSINALDX ($NMLI.$LIBJCL,&RFLI);
COMM '*** PDS400 ***';
STEP PDS400,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
SZ 60;
ASG PACDMB,TMBREST,TEMPRY,&RFTM;
ASG PACDDJ,$NMDJ.$ROOT$FILEDJ,&RFDJ;
ASG PACDBB,SV.BB,FILESTAT=UNCAT,&DVT,&MDT,FSN=ANY;
ASG PACDDC,$NMTU.$ROOT$FILEDC,&RFTU;
ASG PACDDA,$NMTU.$ROOT$FILEDA,&RFTU;
ASG PACDDX,$NMTU.$ROOT$FILEDX,&RFTU;
ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU;
ASG PACDMS,TPACDMS,TEMPRY,&RFTM,END=PASS;
ASG PACDRU,SYS.OUT;
ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PDS450 ***';
STEP PDS450,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
SZ 60;
ASG PACDMS,TPACDMS,TEMPRY,&RFTM;
ASG PACDDA,$NMTU.$ROOT$FILEDA,&RFTU;
ASG PACDDC,$NMTU.$ROOT$FILEDC,&RFTU;
ASG PACDDX,$NMTU.$ROOT$FILEDX,&RFTU;
ASG PACDBJ,$NMBU.$ROOT$FILEBJ,&RFBU;
ASG PACDDE,$NMTU.$ROOT$ROOTDE,&RFTU;
ASG PACDRU,SYS.OUT;
ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND 'DSINDRS1 - ABNORMAL END OF RUN (I/O ERROR)';
LET SEV 3;
END:

```

7. LANCEMENT DU TDS (DSEXTDDS)

Procédure de lancement livrée :

DSEXTDDS : exécution du TDS correspondant au source 'STDSD' (DSMS seul).

Dans la procédure utilisée, le paramètre '&1' doit être positionné à 'STEP1' afin d'effectuer le chargement des TPRs en BACKING STORE (VL=STEP1).

Dans la deuxième étape, il est prévu la réinitialisation automatique des fichiers DS80HE (Sauvegarde de l'écran pour fonction souffleur) et DS80JB fichier relatif nécessaire à la fonction JOB.

LE SOURCE DE LA PROCEDURE

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '* EXECUTION DU TDS : $NMTD *';
COMM '* D S M S *';
COMM '* &1 = BACKING-STORE : STEP1, SINON STEP2 *';
COMM '*****';
MVL STEP1, START='WARM',
    CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
    RFTU=&CTTU$CTTU,
    CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
    RFBS=&CTBS$CTBS,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
    RFBU=&CTBU$CTBU,
    CTDJN='FILESTAT=UNCAT,DVC=$DVDJ,MD=$MDDJ',
    RFDJ=&CTDJ$CTDJ,
    CTTDN='FILESTAT=UNCAT,DVC=$DVTD,MD=$MDTD',
    RFTD=&CTTD$CTTD,
    RFTM='DVC=$DVTM,MD=$MDTM';
JUMP &1;
STEP1:
LIB SM, INLIB1=( $NMTD.$LIBSM, &RFTD);
SYSMAINT COMFILE=*DEMER;
$IN DEMER;
SM;
LOAD MODULE=$TPR0 INPUT=INLIB1 REPLACE;
LOAD MODULE=$TPR1 INPUT=INLIB1 REPLACE;
$EIN DEMER;
STEP2:
IV DSINALJB ($NMLI.$LIBJCL, &RFLI);
IV DSINALHE ($NMLI.$LIBJCL, &RFLI);
JOB LIB SM, $NMTD.$LIBSM;
STEP $NMTD, FILE=( $NMLI.$LIBLM, &RFLI ), DUMP=DATA, OPTIONS=&START;
SZ 150, POOLSZ=70, NBBUF=70;
ASG H_BJRN, DVC=$DVTM, MD=$MDTM, TEMPRY, NEXT, POOL;
DEF H_CTLM, JOURNAL=BEFORE;
ASG DEBUGFILE, $NMTD.DEBUG, &RFTD,
    SHARE=DIR;
ASG DS80IE, $NMTU.$ROOT$ROOTDE,
    &RFTU, ACC=WRITE, SHARE=MONITOR;
DEF DS80IE, JOURNAL=BEFORE;
ASG DS80IA, $NMTU.$ROOT$FILEDA,
    &RFTU, ACC=WRITE, SHARE=MONITOR;
DEF DS80IA, JOURNAL=BEFORE;
ASG PB80DC, $NMTU.$ROOT$FILEDC,
    &RFTU, ACC=WRITE, SHARE=MONITOR;
DEF PB80DC, JOURNAL=BEFORE;
ASG DS80IJ, $NMTU.$ROOT$FILEDJ,
    &RFDJ, ACC=WRITE, SHARE=MONITOR;
DEF DS80IJ, JOURNAL=BEFORE;
ASG DS80IX, $NMTU.$ROOT$FILEDX,
    &RFTU, ACC=WRITE, SHARE=MONITOR;
DEF DS80IX, JOURNAL=BEFORE;
ASG DS80JB, $NMTU.$ROOT$FILEJB,
    &RFTU, ACC=WRITE, SHARE=MONITOR;
DEF DS80JB, JOURNAL=BEFORE;
ASG DS80HE, $NMTU.$ROOT$FILEHE,
```

I N S T A L L A T I O N  
DEROULEMENT DE L'IMPLANTATION

PAGE

213

18  
4

```
        &RFTU,ACC=WRITE,SHARE=MONITOR;  
DEF DS80HE,JOURNAL=BEFORE;  
ASG DS80PA,$NMTU.$ROOT$FILEPA,  
    &RFTU,ACC=WRITE,SHARE=MONITOR;  
DEF DS80PA,JOURNAL=BEFORE;  
ESTP;  
FILLIST IF=($NMTU.$ROOT$FILEDX,&RFTU) USAGE;
```

## 18.5. INTEGRATION DANS UN TDS EXISTANT

### INSTALLATION DANS UN TDS EXISTANT

A l'interprétation du JCL, l'utilisateur prendra soin de vérifier la compatibilité des paramètres DSMS avec les noms existant déjà (par exemple \$NMTD, \$TPR, \$LIBSM, \$LIBLM).

Une fois les JCL obtenus, le schéma d'implantation défini au chapitre précédent reste identique à quelques exceptions près.

#### 1. Allocation des fichiers du système.

Cette étape doit être exécutée telle quelle.

#### 2. Préparation du TDS gérant DSMS.

On n'exécutera pas cette étape mais on vérifiera que la TPRn définie par \$TPR0 et \$TPR1 a bien été initialisée et que la bibliothèque \$NMTD.\$LIBSM est assez grande pour accueillir DSMS (compter 30 cylindres de type MS/D500).

#### 3. Déchargement des fichiers et des programmes.

Cette étape doit être exécutée telle quelle.

#### 4. Génération du TDS gérant DSMS.

En plus de la préparation indiquée au chapitre précédent, l'utilisateur devra fusionner le STDS existant avec le source livré par DSMS (STDSD) en intégrant les clauses USE, SELECT, FD, FILE-INTEGRITY, PROCESSING- MODE, MESSAGE et les zones de WORKING-STORAGE SECTION. Vérifier aussi que la longueur du message (clause "MESSAGE-LENGTH") est supérieure à 6000.

Description détaillée de l'étape :

- Ajouter dans le source du TDS existant les sous-programmes TP DSMS : prendre dans le source TDS DSMS les lignes USE DSCHOI, USE

ZAR980, USE ZAR985 et USE PDTPDF (ce dernier uniquement pour le module DAF).

- Dans FILE-CONTROL, ajouter les assignations des fichiers DSMS : prendre dans le source TDS DSMS, les lignes entre FILE-CONTROL et TDS-FILE-DEFINITION.
- Dans TDS-FILE-DEFINITION, ajouter les clauses FD des fichiers DSMS : prendre dans le source TDS DSMS, les lignes entre TDS-FILE-DEFINITION et PROCESSING-CONTROL.
- Dans PROCESSING-CONTROL, ajouter les file-integrity des fichiers DSMS : prendre dans le source TDS DSMS livré, les lignes entre PROCESSING-CONTROL et WORKING-STORAGE.
- Dans WORKING-STORAGE, ajouter les file-status et les clés relatives des fichiers DSMS, prendre dans le source TDS DSMS, les lignes entre WORKING-STORAGE et TRANSACTION-SECTION (sauf le niveau 01 RELKEY).
- Dans TRANSACTION-SECTION, ajouter la description de la transaction DSMS : prendre dans le source TDS DSMS livré, les lignes entre TRANSACTION-SECTION et la fin.

#### REMARQUES IMPORTANTES:

- Si des noms de fichiers DSMS sont déjà utilisés dans le TDS existant (exemple HE et JB de VA Pac), il faut les nommer différemment dans les lignes à recopier du source TDS DSMS livré.
- Si le TDS existant contient VA Pac, le fichier PB80DC existe déjà (ne pas reprendre dans le source TDS DSMS livré les clauses le concernant).

#### ADAPTATION DU JCL DE GENERATION DU TDS EXISTANT (MGEN)

- Rajouter les sous-programmes TP DSMS (DSCHOI, ZAR980 ZAR985 PDTPDF pour le module DAF ):

sous la ligne LIB CU, INLIB1=(lib CUP du TDS existant), rajouter :  
INLIB2=(lib CUP permanente DSMS);

sous la ligne LMN CU, rajouter :  
MV IL2:DSCHOI;STATUS RESET;  
MV IL2:ZAR980;STATUS RESET;

MV IL2:ZAR985;STATUS RESET;  
MV IL2:PDTPDF;STATUS RESET;

### INSTALLATION DANS UN TDS EXISTANT (SUITE)

5. Link-edit des programmes batch et link-edit des TPRs dans la SMLIB prévue pour DSMS.

Cette étape doit être exécutée telle quelle.

6. Création d'une base DSMS de tests.

Cette étape doit être exécutée telle quelle.

7. Lancement du TDS.

L'utilisateur peut exécuter la procédure DSEXTDDS mais dans ce cas, seuls les fichiers DSMS seront assignés et ouverts. Pour assigner et ouvrir tous les fichiers utilisés par les applications du TDS, il faut fusionner le JCL de lancement existant avec celui (DSEXTDDS) livré avec DSMS en transférant les ordres "ASSIGN" et "DEFINE".

Description détaillée de l'étape :

- Dans le step de SYSMANT, ajouter le load des deux TPRs définies pour DSMS : prendre dans le JCL DSEXTDDS les lignes de load concernées.
- Avant le step de lancement du TDS, ajouter les allocations des fichiers HE et JB de DSMS : prendre dans le JCL DSEXTDDS les 2 lignes INVOKE correspondantes.
- Dans le step de lancement du TDS, ajouter les assignations des fichiers DSMS : prendre dans le JCL DSEXTDDS, les cartes ASSIGN et DEFINE des fichiers suivants : DS80IA, DS80IE, DS80IJ, DS80IX, DS80JB, DS80HE, DS80PA et PB80DC si insertion dans un TDS contenant VisualAge Pacbase, le fichier PB80DC est déjà assigné, ne pas le reprendre).



## 18.6. INSTALLATION DE L'ENVIRONNEMENT DU MODULE DAF

### COMPLEMENT : INSTALLATION ENVIRONNEMENT MODULE 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 dispose de la procédure DPDF pour traiter ses programmes générés utilisant DAF :

- utilisation de la procédure DPDF :
  - . 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 au chapitre "Utilitaires Standard").

#### Sous-programmes et fichiers DAF:

Deux sous-programmes DAF sont fournis à l'installation :

- . PDTPDF pour les requêtes DAF standard,
- . PDSFAC pour les accès physiques à la base DSMS,

Le fichier de travail nécessaire au fonctionnement de DAF TP a un IFN imposé sous TDS : DS80PA. 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 DAFDICF (ou DAFDICA pour la version anglaise) de la librairie 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.

```
COMM 'DSMS 2.5';
COMM '*****';
COMM '*                JCLDAF                *';
COMM '*                =====                *';
COMM '*                EXEMPLE DE JCL D EXECUTION                *';
COMM '*                D UN PROGRAMME DAF BATCH UTILISATEUR                *';
COMM '*****';
MVL  SYSDAF=' $NMTU.$ROOT$FILEPA',USER=' $USER',
     CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     RFTM=' DVC=$DVTM,MD=$MDTM';
IV   DSINALPA,($NMLI.$LIBJCL,&RFLI),
     VL=(USER=&USER,SYSDAF=&SYSDAF);
COMM '*** WITHDAF ***';
STEP WITHDAF,FILE=($NMLI.$LIBLM,&RFLI),REPEAT;
SZ   200;
ASG DS80IE,$NMTU.$ROOT$ROOTDE,&RFTU,
     ACC=READ,SHARE=MONITOR;
DEF DS80IE,READLOCK=STAT;
ASG DS80IA,$NMTU.$ROOT$FILEDA,&RFTU,
     ACC=READ,SHARE=MONITOR;
DEF DS80IA,NBBUF=4;
ASG DS80IX,$NMTU.$ROOT$FILEDX,&RFTU;
ASG PB80DC,$NMTU.$ROOT$FILEDC,&RFTU;
ASG SYSDAF,&SYSDAF&USER,&RFTU;
COMM 'ASG -----';
COMM 'ASG -----';
ESTP;
```

## 18.7. REIMPLANTATION

### PRESENTATION GENERALE

Il est nécessaire de réinstaller le système DSMS lors de la réception d'une sous-version apportant des corrections d'anomalies. Cette sous-version, identifiée par un numéro, est livrée, d'ordinaire sous la forme :

- . d'une bande d'installation 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 bibliothèques de programmes et les fichiers du système sont touchés par cette nouvelle version.

La réinstallation du système DSMS passe par l'exécution de la plupart des procédures utilisées lors de la première installation.

La bande d'installation est décrite dans un sous-chapitre particulier.

Son déroulement est le suivant :

- . Sauvegarde de la bande d'implantation,
- . Réallocation des fichiers système,
- . Déchargement de la bande,
- . Régénération du TDS,
- . Link-edit des programmes,

## DECHARGEMENT DE LA BANDE D'IMPLANTATION

Exécution de la procédure 'DSINUNLD' . Le déchargement de la bande d'implantation est détaillé au Sous-chapitre 'Déroulement de l'Implantation'.

## GENERATION DU TDS

Afin de prendre en compte la nouvelle version des routines DSMS utilisées en TP (DSCHOI, ZAR980 et ZAR985) la génération du TDS est indispensable.

## LINK-EDIT DES PROGRAMMES

Exécution des procédures de link-edit :

- . DSINTLNK link des TPRs,
- . DSINBLNK link des programmes batch.
- . DSINB1LN link des programmes de la procédure DPRT.
- . DSINB2LN link des programmes de la procédure DEXT.
- . DSINB3LN link des programmes de la procédure DUPT.
- . DSINB4LN link des programmes de la procédure DEXH.
- . DSINB5LN link des programmes de la procédure DXBJ.
- . DSINB6LN link des programmes de la procédure DREN.
- . DSINB7LN link du pré-processeur DAF.

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

PAGE

222

18  
7

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
REPRISE DSMS 8.0.1 EN DSMS 2.5 (DR80)

PAGE 223

19

## **19. REPRISE DSMS 8.0.1 EN DSMS 2.5 (DR80)**

## 19.1. 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).



## 19.2. 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 !  
-----
```

## 19.3. REPRISE DE LA SAUVEGARDE

### DR80 : DESCRIPTION DES ETAPES

Cette procédure comprend les étapes suivantes :

.Prise en compte des entrées : CREATE  
.Préparation de conversion : PDSR8B  
.Conversion sauvegarde 8.0.2 : PDSR8C  
.Conversion sauvegarde 1.5 : PDSR15  
.Rotation des générations du fichier d'archive (BB)

PRISE EN COMPTE DES ENTREES : CREATE

Cet utilitaire crée un fichier temporaire à partir d'un fichier défini sous IOF. Il se trouve en tête de toutes les procédures qui possèdent des entrées utilisateur.

. Fichier en entrée : \$INPUT  
. Fichier en sortie (TEMPRY) : TMBRP80  
(reclsize 80, Csize 2048)

### PREPARATION CONVERSION : PDSR8B

.Fichier permanent en entrée :  
-Sauvegarde base DSMS 8.0.1  
Nom externe &OLDBB

.Fichiers de travail en sortie :

fichier temporaire TPACDIQ (reclsize 284)  
fichier temporaire TPACDIT (reclsize 104)  
fichier temporaire TPACDIW (reclsize 92)  
fichier temporaire TPACDLA (reclsize 1)

.Fichiers de tri :

.Fichier mouvement en entrée :  
fichier temporaire TMBDR80

.Etat en sortie :  
-Compte rendu de reprise

CONVERSION SAUVEGARDE 8.0.2 : PDSR8C

.Fichiers de travail en entrée :  
fichier temporaire TPACDIQ (reclsize 284)  
fichier temporaire TPACDIT (reclsize 104)  
fichier temporaire TPACDIW (reclsize 92)  
fichier temporaire TPACDLA (reclsize 1)

.Fichier permanent en entrée :

-Sauvegarde base DSMS 8.0.1  
Nom externe &OLDDB

.Fichier permanent en sortie :  
-Sauvegarde base DSMS convertie  
Nom externe &TPACDB1

.Fichier mouvement en entrée :  
fichier temporaire TMBRP80

.Fichiers de tri :

.Etat en sortie :  
-Compte rendu de reprise  
PAC7IO et PAC7EI

CONVERSION SAUVEGARDE 1.2 : PDSR15

.Fichier permanent en entrée :  
-Sauvegarde base DSMS 1.2 ou 1.5  
Nom externe &TPACDB1

.Fichier permanent en sortie :  
-Sauvegarde base DSMS convertie  
Nom externe \$NMBU.\$ROOT\$FILEBB/G+1

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

ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier archive (BB).

## 19.4. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '*          DSMS      : REPRISE 8.0.1 EN 2.5          *';
COMM '*****';
MVL  PACDBB=' $NMBU.$ROOT$FILEBB', SIZEWK=10, OLDBB=,
      CTTUN=' FILESTAT=UNCAT, DVC=$DVTU, MD=$MDTU',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT, DVC=$DVBS, MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT, DVC=$DVLI, MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT, DVC=$DVBU, MD=$MDBU',
      RFBU=&CTBU$CTBU,
      RFTM=' DVC=$DVTM, MD=$MDTM';
CR   IF=*DR80,
      OF=( TMBRP80, TEMPRY, &RFTM, END=PASS),
      OUTDEF=( CISZ=2048, RECSZ=80, RECFORM=FB);
COMM '*** PDSR8B ***';
STEP PDSR8B, FILE=( $NMLI.$LIBLM, &RFLI), DUMP=DATA;
      SZ 60;
      ASG PACDBB, &OLDBB;
      ASG PACDIQ, TPACDIQ, TEMPRY, &RFTM, END=PASS;
      ASG PACDIT, TPACDIT, TEMPRY, &RFTM, END=PASS;
      ASG PACDIW, TPACDIW, TEMPRY, &RFTM, END=PASS;
      ASG PACDMB, TMBRP80, TEMPRY, &RFTM, END=PASS;
      ASG PACDLA, TPACDLA, TEMPRY, &RFTM, END=PASS;
      ASG PAC7EI, SYS.OUT;
      ASG PACDIK, SYS.OUT;
      SWK WKDISK=( SZ=&SIZEWK, &RFTM);
ESTP;
JUMP ERR, SW20, EQ, 1;
COMM '*** PDSR8C ***';
STEP PDSR8C, FILE=( $NMLI.$LIBLM, &RFLI), DUMP=DATA;
      SZ 60;
      ASG PACDBB, &OLDBB;
      ASG PACDB1, TPACDB1, TEMPRY, &RFTM, END=PASS;
      ALC TPACDB1, UNIT=CYL, SZ=&SIZEWK, INCRSZ=2;
      ASG PACDIQ, TPACDIQ, TEMPRY, &RFTM;
      ASG PACDIT, TPACDIT, TEMPRY, &RFTM;
      ASG PACDIW, TPACDIW, TEMPRY, &RFTM;
      ASG PACDLA, TPACDLA, TEMPRY, &RFTM;
      ASG PACDMB, TMBRP80, TEMPRY, &RFTM, END=PASS;
      ASG PACDIO, SYS.OUT;
      ASG PAC7EI, SYS.OUT;
      SWK WKDISK=( SZ=&SIZEWK, &RFTM);
ESTP;
JUMP ERR, SW20, EQ, 1;
SHIFT $NMBU.$ROOT$FILEBB;
JUMP END;
COMM '*** PDSR15 ***';
STEP PDSR15, FILE=( $NMLI.$LIBLM, &RFLI), DUMP=DATA;
      SZ 100;
      ASG PACDBB, &TPACDB1, TEMPRY, &RFTM;
      ASG PACDB1, &PACDBB/G+1, &RFBU;
      ASG PACDIK, SYS.OUT;
ESTP;
JUMP ERR, SW20, EQ, 1;
SHIFT &PACDBB;
JUMP END;
ERR:
SEND ' DSINDR80 - ABNORMAL END OF RUN (I/O ERROR) ';
```

REPRISE DSMS 8.0.1 EN DSMS 2.5  
JCL DE LA PROCEDURE

(DR80)

PAGE

229

19  
4

LET SEV 3;  
END:

REPRISE DSMS 8.0.1 EN DSMS 2.5  
JCL DE LA PROCEDURE

(DR80)

PAGE

230

19  
4

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
REPRISE DSMS 8.0.2 01/02 EN DSMS 2.5 (DR8Q)

PAGE 231

20

## **20. REPRISE DSMS 8.0.2 01/02 EN DSMS 2.5 (DR8Q)**

## 20.1. 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.



## 20.2. REPRISE DE LA SAUVEGARDE

### DR8Q : DESCRIPTION DES ETAPES

REPRISE DES REQUETES : PDSR8Q

.Fichier permanent en entrée :  
-Sauvegarde base DSMS 8.0.2 01/02  
Nom externe &OLDBB

.Fichier de travail en sortie :  
fichier temporaire TPACDIQ (reclsize 284)

.Fichiers de tri :

FUSION : PDSR8R

.Fichier de travail en entrée :  
fichier temporaire TPACDIQ (reclsize 284)

.Fichier permanent en entrée :  
-Sauvegarde base DSMS 8.0.2 01/02  
Nom externe &OLDBB

.Fichier permanent en sortie :  
-Sauvegarde base DSMS reprise  
Nom externe &TPACDB1

CONVERSION SAUVEGARDE 1.2 : PDSR15

.Fichier permanent en entrée :  
-Sauvegarde base DSMS 1.2 ou 1.5  
Nom externe &TPACDB1

.Fichier permanent en sortie :  
-Sauvegarde base DSMS convertie  
Nom externe \$NMBU.\$ROOT\$FILEBB/G+1

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

ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier archive (BB).

## 20.3. JCL DE LA PROCEDURE

```

COMM '*****';
COMM '*          DSMS      : REPRISE 8.0.2 01          *';
COMM '*****';
MVL  PACDBB=' $NMBU.$ROOT$FILEBB', OLDBB=' BB802', SIZEWK=10,
      CTTUN=' FILESTAT=UNCAT, DVC=$DVTU, MD=$MDTU',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT, DVC=$DVBS, MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT, DVC=$DVLI, MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT, DVC=$DVBU, MD=$MDBU',
      RFBU=&CTBU$CTBU,
      RFTM=' DVC=$DVTM, MD=$MDTM';
COMM '*** PDSR8Q ***';
STEP PDSR8Q, FILE=( $NMLI.$LIBLM, &RFLI), DUMP=DATA;
      SZ 60;
      ASG PACDBB, &OLDBB;
      ASG PACDIQ, TPACDIQ, TEMPRY, &RFTM, END=PASS;
      ASG PAC7EI, SYS.OUT;
      SWK WKDISK=( SZ=&SIZEWK, &RFTM);
ESTP;
JUMP ERR, SW20, EQ, 1;
COMM '*** PDSR8R ***';
STEP PDSR8R, FILE=( $NMLI.$LIBLM, &RFLI), DUMP=DATA;
      SZ 60;
      ASG PACDBB, &OLDBB;
      ASG PACDB1, TPACDB1, TEMPRY, &RFTM, END=PASS;
      ASG PACDIQ, TPACDIQ, TEMPRY, &RFTM;
      ASG PAC7EI, SYS.OUT;
      SWK WKDISK=( SZ=&SIZEWK, &RFTM);
ESTP;
JUMP ERR, SW20, EQ, 1;
COMM '*** PDSR15 ***';
STEP PDSR15, FILE=( $NMLI.$LIBLM, &RFLI), DUMP=DATA;
      SZ 100;
      ASG PACDBB, TPACDB1, TEMPRY, &RFTM;
      ASG PACDB1, &PACDBB/G+1;
      ASG PACDIK, SYS.OUT;
ESTP;
JUMP ERR, SW20, EQ, 1;
SHIFT &PACDBB;
JUMP END;
ERR:
SEND ' DSINDR8Q - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:

```

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
REPRISE DSMS 8.0.2 COMPATIBLE AVEC VA PAC 8.0.1

PAGE 235

21

## **21. REPRISE DSMS 8.0.2 COMPATIBLE AVEC VA PAC 8.0.1**

## 21.1. PRESENTATION

### INTRODUCTION

La version nouvelle de DSMS n'est pas compatible avec les systèmes VisualAge Pacbase version 8.0.1.

Si votre DSMS 8.0.2 était utilisé avec une version 8.0.1 de VA Pac, l'installation de la nouvelle version nécessite de procéder aux opérations suivantes:

- 1.Sauvegarde de la base DSMS (DSAV),
- 2.Implantation de la nouvelle version de DSMS,
- 3.Implantation de la dernière version disponible de VA Pac,
- 4.Reprise du fichier BB de la sauvegarde (procédure DR8X),
- 5.Réorganisation de la base DSMS pour reconstituer le fichier DX,
- 6.Restauration de la base DSMS.

### CONDITION D'EXECUTION

Aucune.

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

## 21.2. PROCEDURE 'DR8X' : ENTREE UTILISATEUR

### DR8X : ENTREE UTILISATEUR

L'entrée utilisateur permet la transposition des codes produit. Elle est composée de 1 à n lignes, chaque ligne étant décomposée en groupes de 6 caractères, à partir de la première colonne.

```
-----  
!Pos.!Lon.! Valeur ! Signification !  
!-----!  
! 1 ! 3 ! 'PRO' ! Ancien code produit !  
! 4 ! 3 ! 'PRO' ! Nouveau code produit !  
! ! ! ! Chaque groupe de 6 caractères peut !  
! ! ! ! être répété 13 fois maximum par ligne !  
-----
```

## 21.3. PROCEDURE 'DR8X'

### DR8X : DESCRIPTION DES ETAPES

#### CONVERSION POUR VA PAC 8.0.2 : PDSR8X

.Fichier permanent en entrée :  
-Sauvegarde base DSMS 8.0.1  
  Nom externe &OLDBB

.Fichiers de tri

.Fichier mouvement en entrée :  
  fichier temporaire TMBRPDC

.Fichier permanent en sortie :  
-Sauvegarde base DSMS convertie  
  Nom externe &TPACDB1

.Etat en sortie  
-Compte rendu de reprise  
  PACDIK et PACDEI pour les anomalies

#### CONVERSION SAUVEGARDE 1.2 : PDSR15

.Fichier permanent en entrée :  
-Sauvegarde base DSMS 1.2 ou 1.5  
  Nom externe &TPACDB1

.Fichier permanent en sortie :  
-Sauvegarde base DSMS convertie  
  Nom externe \$NMBU.\$ROOT\$FILEBB/G+1

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

#### ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier archive (BB).

## 21.4. PROCEDURE 'DR8X' : JCL

```
MVL  PACDBB=' $NMBU.$ROOT$FILEBB',SIZEWK=10,OLDBB=,
      CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      RFBU=&CTBU$CTBU,
      RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*DR8X,
      OF=(TMBRPDC,TEMPRY,&RFTM,END=PASS),
      OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PDSR8X ***';
STEP PDSR8X,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
      SZ 60;
      ASG PACDBB,&OLDBB;
      ASG PACDMB,TMBRPDC,TEMPRY,&RFTM,END=PASS;
      ASG PACDB1,TPACDB1,TEMPRY,&RFTM,END=PASS;
      ALC TPACDB1,UNIT=CYL,SZ=&SIZEWK,INCRSZ=2;
      ASG PAC7EI,SYS.OUT;
      ASG PACDIK,SYS.OUT;
      SWK WKDISK=(SZ=&SIZEWK,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
SHIFT $NMBU.$ROOT$FILEBB;
JUMP END;
COMM '*** PDSR15 ***';
STEP PDSR15,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
      SZ 100;
      ASG PACDBB,TPACDB1,TEMPRY,&RFTM;
      ASG PACDB1,&PACDBB/G+1,&RFBU;
      ASG PACDIK,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
SHIFT &PACDBB;
JUMP END;
ERR:
SEND ' DSINDR8X - ABNORMAL END OF RUN (I/O ERROR) ';
END:
```

REPRISE DSMS 8.0.2 COMPATIBLE AVEC VA PAC 8.0.1  
PROCEDURE 'DR8X' : JCL

PAGE

240

21  
4



VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
REPRISE DSMS 1.2 ou 1.5 en DSMS 2.5

PAGE 241

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 : Nom externe &OLDBB

.Fichier permanent en sortie :  
-Sauvegarde base DSMS reprise  
PACDB1 : Nom externe \$NMBU.\$ROOT\$FILEBB/G+1

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

ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier archive (BB).

## 22.3. PROCEDURE 'DR15' : JCL

```
COMM '*****';
COMM '* REPRISE DE LA BASE DSMS 1.2 OU 1.5 *';
COMM '*****';
MVL OLDBB='BB1215',
    PACDBB='$NMBU.$ROOT$FILEBB',SIZEWK=10,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
COMM '*** PDSR15 ***';
STEP PDSR15,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
    SZ 100;
    ASG PACDBB,&OLDBB;
    ASG PACDB1,&PACDBB/G+1;
    ASG PACDIK,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
SHIFT &PACDBB;
JUMP END;
ERR:
SEND ' DSINDR15 - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:
```

## 22.4. PROCEDURE 'DR5J' - DESCRIPTION DES ETAPES

REPRISE DU JOURNAL: PDSR5J

.Fichier permanent en entrée :  
-Journal séquentiel 1.2 ou 1.5  
PACDBJ : Nom externe &OLDBJ

.Fichier permanent en sortie :  
-Journal repris au format 2.5  
PACDJB : Nom externe \$NMBU.\$ROOT\$FILEBJ/G+1

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

ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier archive (BJ).

## 22.5. PROCEDURE 'DR5J' : JCL

```
COMM '*****';
COMM '* REPRISE DU JOURNAL DSMS 1.2 OU 1.5 *';
COMM '*****';
MVL OLDBJ='BJ1215',
    PACDBJ='$NMBU.$ROOT$FILEBJ',SIZEWK=2,
    CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
    RFLI=&CTLI$CTLI,
    RFTM='DVC=$DVTM,MD=$MDTM';
COMM '*** PDSR5J ***';
STEP PDSR5J,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA;
    SZ 100;
    ASG PACDBJ,&OLDBJ;
    ASG PACDJB,&PACDBJ/G+1;
    ASG PACDIK,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
SHIFT &PACDBJ;
JUMP END;
ERR:
SEND ' DSINDR5J - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:
```

VISUALAGE PACBASE - MANUEL D'EXPLOITATION  
DSMS - EXPLOITATION & INSTALLATION  
REPLACEMENT DES LOW-VALUES PAR DES BLANCS (DLVB)

PAGE 247

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

.Indiquer &OPTS='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 : Nom externe \$NMBU.\$ROOT\$FILEBB

.Fichier en sortie :

-Sauvegarde de la base

PACDB1 : Nom externe \$NMBU.\$ROOT\$FILEBB/G+1

ROTATION DES GENERATIONS DU FICHER ARCHIVE : SHIFT

Cette étape effectue une rotation des générations du fichier  
archive (BB).

### 23.3. DLVB : PROCEDURE D'EXECUTION

```
COMM '*****';
COMM '*          DSMS : SUPPRESSION DES LOW-VALUES DANS BB *';
COMM '*                                           *';
COMM '*          DSEXDLVB                                           *';
COMM '*          =====                                           *';
COMM '*                                           *';
COMM '*****';
MVL  PACDBB=' $NMBU.$ROOT$FILEBB',OPTS=,
      CTTUN=' FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
      RFTU=&CTTU$CTTU,
      CTBSN=' FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
      RFBS=&CTBS$CTBS,
      CTLIN=' FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
      RFLI=&CTLI$CTLI,
      CTBUN=' FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
      RFBU=&CTBU$CTBU,
      RFTM=' DVC=$DVTM,MD=$MDTM';
COMM '*** PDSLVB ***';
STEP PDSLVB,FILE=( $NMLI.$LIBLM,&RFLI),DUMP=DATA,
      OPTIONS=' '&OPTS'';
      SZ 100;
      ASG PACDBB,&PACDBB,&RFBU;
      ASG PACDB1,&PACDBB/G+1,&RFBU;
      ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP ERR,SW30,EQ,1;
SHIFT $NMBU.$ROOT$FILEBB;
JUMP END;
ERR:
SEND ' DSINDLVB - ABNORMAL END OF RUN (I/O ERROR) ';
LET SEV 3;
END:
```