



VisualAge Pacbase 2.5

**DSMS 2.5 IBM MVS/IMS
OPERATIONS MANUAL**

DEDIM000252A

Note

Before using this document, read the general information under "Notices" on the next page.

According to your license agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:

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

Consult the Catalog section in the Documentation home page to make sure you have the most recent edition of this document.

Second Edition (August 1999)

This edition applies to the following licensed program:

- VisualAge Pacbase Version 2.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at:

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

or to the following postal address:

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

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1983, 1999. All rights reserved.

Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

NOTICES

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of information which has been exchanged, should contact:

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

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM may change this publication, the product described herein, or both.

TRADEMARKS

IBM is a trademark of 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, and VisualAge are trademarks of International Business Machines Corporation, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

All other company, product, and service names may be trademarks of their respective owners.

TABLE OF CONTENTS

1. FOREWORD	9
2. DSMS COMPONENTS	11
2.1. INTRODUCTION	12
2.2. SYSTEM-PARAMETERS LIBRARY.....	13
2.3. THE ON-LINE PROGRAMS LIBRARY	15
2.4. THE BATCH PROGRAMS LIBRARY	17
2.5. LIBRARIES SPECIFIC TO IMS	20
2.6. THE SYSTEM FILES	21
2.7. THE USER FILES.....	22
2.8. THE LIBRARY OF ENTRY-POINT SOURCES	25
3. ENVIRONMENT	26
3.1. ORGANIZATION OF DL/1 DATABASES	27
3.2. SYSTEM SPACE REQUIREMENTS.....	31
4. BATCH PROCEDURES	32
4.1. INTRODUCTION	33
4.2. CLASSIFICATION OF PROCEDURES	34
4.3. ABNORMAL EXECUTION.....	36
5. JOURNAL ARCHIVING (DARC)	37
5.1. INTRODUCTION	38
5.2. INPUT - PROCESSING - RESULTS	39
5.3. DESCRIPTION OF STEPS.....	42
5.4. EXECUTION JCL.....	44
6. PRINTING OF QUERIES AND OUTPUT REPORTS (DPRT)	47
6.1. INTRODUCTION	48
6.2. INPUT - PROCESSING - RESULTS	49
6.3. DESCRIPTION OF STEPS.....	52
6.4. EXECUTION JCL.....	53
7. DATABASE RESTORATION (DRST)	55
7.1. INTRODUCTION	56
7.2. INPUT - PROCESSING - RESULTS	58
7.3. DESCRIPTION OF STEPS.....	60
7.4. EXECUTION JCL.....	63
8. DATABASE BACKUP (DSAV)	66
8.1. INTRODUCTION	67
8.2. INPUT - PROCESSING - RESULTS	68
8.3. DESCRIPTION OF STEPS.....	69
8.4. EXECUTION JCL.....	70
9. REORGANIZATION OF CROSS-REFERENCE FILE (DREO)	72
9.1. INTRODUCTION	73
9.2. INPUT - PROCESSING - RESULTS	74
9.3. DESCRIPTION OF STEPS.....	75
9.4. EXECUTION JCL.....	77
10. EXTRACTION FROM VA PAC ARCHIVED JOURNAL (DEXP)	80
10.1. INTRODUCTION	81
10.2. INPUT - PROCESSING - RESULTS	82
10.3. DESCRIPTION OF STEPS.....	84
10.4. EXECUTION JCL.....	85

11. EXTRACTION OF ENTITIES	(DEXT)	87
11.1. INTRODUCTION		88
11.2. INPUT - PROCESSING - RESULTS		89
11.3. DESCRIPTION OF STEPS		92
11.4. EXECUTION JCL		93
12. EXTRACTION OF TABLES FOR EXTERNAL LISTS	(DEXH)	95
12.1. INTRODUCTION		96
12.2. INPUT - PROCESSING - RESULTS		97
12.3. DESCRIPTION OF STEPS		98
12.4. EXECUTION JCL		99
13. BATCH UPDATE OF ENTITIES	(DUPT)	101
13.1. INTRODUCTION		102
13.2. INPUT - PROCESSING - RESULTS		104
13.3. DESCRIPTION OF STEPS		110
13.4. EXECUTION JCL		111
14. FILE INITIALIZATION	(DINI)	113
14.1. INTRODUCTION		114
14.2. INPUT - PROCESSING - RESULTS		115
14.3. DESCRIPTION OF STEPS		117
14.4. EXECUTION JCL		118
15. JOURNAL EXTRACTION FOR UPDATE	(DXBJ)	119
15.1. INTRODUCTION		120
15.2. INPUT - PROCESSING - RESULTS		121
15.3. DESCRIPTION OF STEPS		122
15.4. EXECUTION JCL		123
16. CODE AND KEYWORD UPDATE	(DREN)	125
16.1. INTRODUCTION		126
16.2. INPUT - PROCESSING - RESULTS		127
16.3. DESCRIPTION OF STEPS		130
16.4. EXECUTION JCL		131
17. PRE-PROCESSING OF GENERATED DAF PROGRAMS	(DPDF)	133
17.1. INTRODUCTION		134
17.2. INPUT - PROCESSING - RESULTS		135
17.3. DESCRIPTION OF STEPS		136
17.4. EXECUTION JCL		137
18. BATCH UPDATE FROM DAF TABLES	(DUPD)	139
18.1. INTRODUCTION		140
18.2. INPUT - PROCESSING - RESULT		141
18.3. DESCRIPTION OF STEPS		142
18.4. EXECUTION JCL		144
19. INSTALLATION		146
19.1. ENVIRONMENT PREPARATION		147
19.2. INTRODUCTION		149
19.3. INSTALLATION CARDRIDGE (OR TAPE)		150
19.4. INSTALLATION PREPARATION		151
19.4.1. INITIAL JCL		153
19.4.2. COMPLETE JCL INSTALLATION		158
19.4.3. INSTALLATION DEFAULT OPTIONS		162
19.4.4. INSTALLATION VARIANTS		163
19.4.5. JCL MODULES		164
19.4.6. JCL PARAMETERS		166
19.4.7. JCL MODULES SEPARATORS		170

19.5. INSTALLATION PROCESS	171
19.5.1. LOADING BATCH AND ON-LINE PROGRAMS	172
19.5.2. ALLOCATION/LOADING OF SY PARAMETERS PDS.....	177
19.5.3. LOADING OF BATCH OPERATION PROCEDURES	185
19.5.4. LOADING OF THE DBDLIB	187
19.5.5. LOADING OF THE PSBLIB	189
19.5.6. INITIAL PREPARATION OF FILES.....	191
19.5.7. LOADING OF THE ERROR MESSAGE FILE.....	194
19.5.8. COMPILATION OF ACBs.....	196
19.5.9. COMPLEMENT: ENTRY-POINT SOURCES FOR USER CHECKS.....	198
19.5.10. LOADING OF THE DSMS TEST DATABASE	203
19.6. LIST OF INSTALLED PROGRAMS	205
19.7. USE TESTS	208
19.7.1. TEST JCL: DEXT.....	210
19.7.2. TEST JCL: DARC	211
19.7.3. TEST JCL: DSAV.....	212
19.7.4. TEST JCL: DRST.....	213
19.7.5. TEST JCL: DXBJ.....	214
19.8. SYSTEM REINSTALLATION.....	215
20. RETRIEVAL OF DSMS 8.0.1 --> DSMS 2.5 (DR80).....	218
20.1. OPERATIONS TO CARRY OUT	219
20.2. USER INPUT	220
20.3. BACKUP RETRIEVAL.....	221
20.4. EXECUTION JCL.....	223
21. RETRIEVAL OF DSMS 8.0.2 01/02 --> DSMS 2.5 (DR8Q).....	225
21.1. OPERATIONS TO CARRY OUT	226
21.2. BACKUP RETRIEVAL.....	227
21.3. EXECUTION JCL.....	228
22. RETRIEVAL OF DSMS 1.2 OR 1.5 --> DSMS 2.5	229
22.1. OPERATIONS TO CARRY OUT	230
22.2. 'DR15' PROCEDURE - DESCRIPTION OF STEPS	231
22.3. 'DR15' PROCEDURE - EXECUTION JCL	232
22.4. 'DR5J' PROCEDURE - DESCRIPTION OF STEPS	233
22.5. 'DR5J' PROCEDURE - EXECUTION JCL.....	234
23. REPLACEMENT OF LOW-VALUES BY BLANKS (DLVB)	235
23.1. DLVB: REPLACEMENT OF LOW-VALUES BY BLANKS	236
23.2. DLVB: PARAMETERS-DESCRIPTION OF STEPS	237
23.3. DLVB: EXECUTION JCL.....	238

VisualAge Pacbase - Operations Manual	PAGE	9
DSMS - INSTALLATION & OPERATIONS		
FOREWORD		1

1. FOREWORD

FOREWORD

USE OF THE MANUAL

This manual is intended for the person in charge of the installation and for the DSMS Database Manager.

It describes the DSMS components, the environment, the batch procedures, the instructions for installing the new version and the operations to be carried out for a standard reinstallation of corrected versions.

NOTE

DSMS 2.5 requires a complete installation of the technical package, i.e. files, programs and batch procedures.

SITES WITH FORMER RELEASES

Once the installation is complete, refer to the relevant chapter for the retrieval of the site's previous release:

- Retrieval of DSMS 8.0.1 and adaptation to DSMS 2.5,
- Retrieval of DSMS 8.0.2 01 or 02 and adaptation to DSMS 2.5,
- Retrieval of DSMS 8.0.2 01 or 02 compatible with VA Pac 8.0.1 and adaptation to DSMS 2.5,
- Retrieval of DSMS 1.2 or 1.5 and adaptation to DSMS 2.5,

and follow the instructions carefully in order to ensure compatibility between the new release and the former one.

VisualAge Pacbase - Operations Manual	PAGE	11
DSMS - INSTALLATION & OPERATIONS		
DSMS COMPONENTS		2

2. DSMS COMPONENTS

2.1. INTRODUCTION

INTRODUCTION

DSMS manages permanent data in batch and on-line mode.

Three types of resources are required for the operation of DSMS.

- Libraries which store the DSMS operating programs and system parameters:
 - . An on-line program library
 - . A batch program library

- Permanent files containing data manipulated by the DSMS system programs previously defined:
 - . A system file containing error messages and HELP documentation on DSMS,
 - . User files containing the User and Administrator data.

- A library containing the operations parameters.

NOTE: This manual describes the installation and operation of DSMS. DSMS can be installed independently of other VisualAge Pacbase functions and facilities.

IMPORTANT: For the IMS platform, the VA Pac element file (DC-D3) is defined when installing the VisualAge Pacbase System.

For further details on the operation of the Function itself, refer to the DSMS Reference Manual.

2.2. SYSTEM-PARAMETERS LIBRARY

THE SYSTEM PARAMETERS LIBRARY: SY

Its required size is approximately 9 tracks on a 3390 disk.

It is a PDS file which contains the entries of the utilities used in the batch installation and operation procedures, as well as the DBD and PSB sources.

.DL/1 source files:

```
+-----+-----+
! Members  ! Contents                                     !
+-----+-----+
! PACDwwxx ! DBD of database ww, xx suffix                !
! Ppppppyy ! PSB of batch program, yy suffix             !
! rr00zz   ! rr: root of on-line monitor zz's PSB       !
+-----+-----+

+-----+-----+
! MEMBERS  ! CONTENTS : DELETE/DEFINE OF THE FILE        !
+-----+-----+
! DFxyyDA  ! DSMS data (DA)                              !
! DFxyyDl  !                                             !
! DLxyyDJ  ! DSMS journal (DJ)                          !
! DFxyyDX  ! Cross-references (DX)                      !
! DFxxxxDE ! Error messages and HELP documentation (DE) !
! DFxxxxDZ ! SPAs database (DZ)                        !
! DFxxxxDF ! On-line DAF work file                    !
! DFxxxxF1 ! - - - - -                                 !
+-----+-----+
```

See the file physical characteristics in the following sub-chapters.

xx = ROOT System root
yy = FILE DSMS user database number

The information concerning the catalogue in use, the disks, the blocking factor, etc., is initialized according to the initial installation parameters and can be modified if need be by the DSMS Databse Manager.

.The VERIFY's, REPRO's and LISTCAT's of DSMS files:

- The VERIFF members (ff = DA, DC, DJ, DX,...) contain the VERIFY PACDff order for each one of the DSMS files.
- The REPROff members (ff = DZ, DF) contain the REPRO INFILE (INzz) OUTFILE (OUTzz) order.
- The LIXxyyDJ member contains the LISTCAT of the DJ Journal file.

.The DFSVSAMP's:

- The DFSVSAMn(n=8,9 ou M) members contains the input for control cards of the VSAM bufferization. These cards are initialized with common values during system installation, but their management is the responsibility of the product's System Manager.

Parameters defined under the name DFSVSAMn, with n=M, allow for optimization of the batch update times when executing the corresponding procedure.

.Records for initialization of DZ and DF:

- In members LDxxyyDF and LDxxyyDZ.

.APPLCTN and TRANSACT Macro-instructions

- The PACSCTRL member contains all the macro-instructions which must be defined in the IMS Control Region. This PDS is used by the system operations staff.

NOTE

All modifications of file characteristics must be executed in the System-Parameter Library.

2.3. THE ON-LINE PROGRAMS LIBRARY

ON-LINE PROGRAMS LIBRARY

Its size is approximately 70 tracks of a 3390 disk.

```
+-----+
! PROGRAM ! CORRESPONDING CHOICE !
! CODE    ! COMMENTS                      !
+-----+
! xxCHOI  ! CHOICE decoding sub-program    !
! xxCUAM  ! suppl. check sub-program on chge !
! xxCUEV  ! suppl. check sub-program on event !
! xxCUMQ  ! suppl. check sub-program on rept !
! xxCURQ  ! suppl. check sub-program on query !
! xxCUSI  ! suppl. check sub-program on site !
! xxR980  ! 3270 message                    !
! xx0099  ! DSMS Monitor                     !
! xx00ZZ  ! DSMS Monitor                     !
! xx00AA  ! Initial screen                   !
! xx00AB  ! Abend Map                         !
! 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  ! On-line HELP function            !
! xx00JO  ! JO                                !
! xx00KA  ! HK                                !
! xx00K1  ! LGK_ LAK_                        !
! xx00K2  ! LPK_                              !
! xx00K3  ! WS  WU                            !
! xx00LE  ! LDE_ LNE_ LSE_ LDC LNC          !
! xx00LS  ! LIE .....*...                    !
! xx00MA  ! H (General Menu)                 !
! xx00PA  ! HP                                !
! xx00P1  ! PL                                !
+-----+
```

PROGRAM	CORRESPONDING CHOICE
CODE	COMMENTS
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
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 'word-processor' ! ! information !

PROGRAM	DESCRIPTION
CODE	
PACSECB	Security Systems Interface
PACX10	Lowercase management
PACXAB	ABEND management
MVSJOB	MVS 'JOB' function

NOTE

'xx' is the program prefix, corresponding to the ROOT parameter (first 2 characters of the chosen transaction code).

2.4. THE BATCH PROGRAMS LIBRARY

BATCH PROGRAMS LIBRARY: MBR8

Its size is approximately 55 tracks of a 3370 disk.

! CODE	! PROC	! COMMENTS	!
! DAFD10	! DPDF	! DAF pre-processor	!
! PACSECB	!	! Security Systems Interface	!
! PDCHOI	! DUPT	! Sub-program	!
! PDSA10	! DPRT	! DPRT print sub-program	!
! PDSB	! -	! DPRT flow monitor (French)	!
! PDSBE	! -	! DPRT flow monitor (English)	!
! PDSBAS	! DSAV	! Checks data integrity	!
! PDSCAM	! DUPT	! Suppl. check sub-program for PDSUB1	!
! PDSCEV	! -	! Suppl. check sub-program for PDSUE1	!
! PDSCMQ	! -	! Suppl. check sub-program for PDSUQ6	!
! PDSCRQ	! -	! Suppl. check sub-program for PDSUQ1	!
! PDSCSI	! -	! Suppl. check sub-program for PDSUS1	!
! PDTPDF	! -	! Extraction sub-program for DAF	!
! PDSEX	! DEXT	! DEXT flow monitor (french)	!
! PDSEXE	! DEXT	! DEXT flow monitor (english)	!
! PDSE90	! DPRT	! DPRT print sub-program	!
! PDSERQ	! DPRT	! DPRT print sub-program (request/layout)	!
! PDSFAC	! -	! File-access sub-program	!
! PDSINI	! DINI	! Initializes DSMS files	!
! PDSJMS	! DREN	! Changes the codes in the Journal	!
! PDSLVB	! DLVB	! Replaces low-values by blanks in the BB	!
!	! -	! backup file	!
! PDSMS	! DREN	! Replacement monitor for table codes,	!
!	! -	! keywords and site codes	!
! PDSMSE	! -	! Same as PDSMS (english version)	!
! PDSRCT	! DREN	! Input transaction check	!
! PDSRFU	! -	! Sorts merges	!
! PDSRMS	! -	! Changes the codes in the backup	!
! PDSRQ0	! DPRT	! Analyzes requests	!
! PDSRQ1	! -	! Selects and extracts requests	!
! PDSRQ2	! -	! Formats elements	!
! PDSRQ3	! -	! Extracts and prints the data	!
! PDSR10	! DREO	! Reorganizes the cross-reference file	!
! PDSR20	! -	!	!
! PDSR30	! -	!	!
! PDSR40	! -	!	!
! PDSUAA	! DUPT	! Sub-program	!

! CODE	! PROC	! COMMENTS
! PDSUB1	! -	! -
! PDSUB2	! -	! -
! PDSUB3	! -	! -
! PDSUB4	! -	! -
! PDSUE1	! -	! -
! PDSUE2	! -	! -
! PDSUE3	! DUPT	! Sub-program
! PDSUK1	! -	! -
! PDSUP0	! -	! Monitor
! PDSUP1	! -	! Sub-program
! PDSUQ1	! -	! -
! PDSUQ2	! -	! -
! PDSUQ5	! -	! -
! PDSUQ6	! -	! -
! PDSUQ7	! -	! -
! PDSUQ8	! -	! -
! PDSUS1	! -	! -
! PDSUS3	! -	! -
! PDSUS4	! -	! -
! PDSUS6	! -	! -
! PDSUTT	! -	! -
! PDSUTV	! -	! -
! PDSUTW	! -	! -
! PDSUTX	! -	! -
! PDSUTY	! -	! -
! PDSUTZ	! -	! -
! PDSUT1	! -	! -
! PDSUT2	! -	! -
! PDSUT3	! -	! -
! PDSUT4	! -	! -
! PDSUT5	! -	! -
! PDSUT6	! -	! -
! PDSUT7	! -	! -
! PDSUT8	! -	! -
! PDSUT9	! -	! -
! PDSXCT	! DEXT	! Checks validity of input
! PDSXDT	! DINS	! Lists the installed programs
! PDSXST	! DEXT	! Sorts entities
! PDSXTH	! DEXH	! Extracts tables to create external lists!
! PDSXTR	! -	! Extracts entities
! PDS300	! DARC	! Archives and deactivates the journal
! PDS320	! -	! Reinitializes the journal

```

+-----+
! CODE   ! PROC ! COMMENTS !
+-----+
! PDS380 ! DRST ! Verifies the journal !
! PDS400 ! -    ! Restores or initializes the files !
! PDS450 ! -    ! Retrieves the archived journal !
! PDS500 ! DSAV ! Saves data/elements/references !
! PDS600 ! DEXP ! Extracts VA Pac journal !
! PDS610 ! -    ! !
! REP2PJ ! DEXQ ! Extracts VA Pac journal < 2.0 !
! PDS700 ! DXBJ ! Extracts journalized transactions !
! PDS900 ! DUPD ! Updates DSMS - DAF database !
! PTU001 ! All  ! Copies entries onto a disk file !
! PDSR8B ! DR80 ! Retrieval of an 8.0 / 8.01 DSMS database!
! PDSR8C !     ! IMPORTANT: VA PAC MUST HAVE PREVIOUSLY !
!     !     ! BEEN UPDATED TO VERSION 8.0.2. !
+-----+
! PDSR8Q ! DR8Q ! Retrieval of a DSMS 8.0.2 01/02 Database!
! PDSR8R !     ! (Retrieval of Queries) !
+-----+
! PDSR15 ! DR15 ! Retrieval of a DSMS 1.2/1.5 database !
! PDSR5J ! DR5J ! Retrieval of archived journal DSMS 1.5 !
+-----+
! PDSV10 ! DLDE ! Loading of error-message database !
! PDSV20 ! DLDZ ! Loading of S.P.A. database !
! PTV090 ! LDDF ! Loading of DAF file database !
+-----+

```

SECURITY SYSTEMS INTERFACE EXTENSION

This extension contains sub-programs which allow DSMS to connect to the security system specific to the site.

For RACF, the PACSECU8 sub-program must be installed in an authorized library, by copying the module located in the batch module library (PACD.MBR8).

```

+-----+
! PROGR. ! Renamed ! Security system !
+-----+
! PACSECRA ! PACSECU8 ! RACF !
! PACTSS ! ! TOPSECRET Batch !
+-----+

```

For details on how to use this extension, refer to the SECURITY SYSTEMS INTERFACE Reference Manual.

2.5. LIBRARIES SPECIFIC TO IMS

LIBRARIES SPECIFIC TO IMS

There are two libraries specific to IMS:

- . The library of DBDs (DBDLIB),
- . The library of PSBs (PSBLIB),

If you do not use existing system libraries, you will have to create them according to the following requirements:

- . Organization : PDS
- . DSN : to be chosen at installation
- . DCB : RECFM=U,BLKSIZE=6144
- . Size : See Chapter 'ENVIRONMENT'

2.6. THE SYSTEM FILES

SYSTEM FILES

They make up the actual system. They are not affected by daily transactions, and they must be reloaded each time the system is reinstalled.

They are the LIBRARIES described in the previous subchapters:

- .The library of on-line executable modules MTR8,
- .The library of batch executable modules MBR8,
- .The library of system parameters SY,

as well as the file containing the ERROR MESSAGES and the HELP DOCUMENTATION of the DSMS function (DE):

.Size : Approximately 30,000 records
.Organization : DL/1 HISAM database
.SEGM length : 90
.RECORD length: 98
.Key : 17 (position 0)
.DSN : &INDEX.&ROOT.&ROOT.DE
.DBD : PACDDExx

2.7. THE USER FILES

USER FILES

These files contain user information which are managed by the DSMS function.

The first six (4+2 indexes) contain the data directly managed by this function.
They are:

.The DSMS Data file (DA)

```
.Organization : DL/1 HIDAM VSAM database
.SEGM. length : 80 minimum, 350 maximum
.RECORD length: 4,096
.DSN          : &INDEX..&ROOT.&FILE.DA
.DBD          : PACDDAxx
.Size         : depending on records' lengths
```

.The primary index of data (D1)

```
.Organization : INDEX/VSAM 'DA' index
.SEGM. length : 40 bytes
.RECORD length: 59 bytes
.DSN          : &INDEX..&ROOT.&FILE.D1
.DBD          : PACDD1xx
.Size         : 22 records per 1024-CI
```

.The VA Pac Elements file (DC)

```
.Organization : DL/1 HIDAM/VSAM database
.SEGM. length : 50 minimum, 168 maximum
.RECORD length: 4,096 bytes
.DSN          : &INDEX..&ROOT.&FILE.DC
.DBD          : PACDDCxx
.Size         : depending on records' lengths
```

.The primary index of VA Pac elements (D3)

.Organization : Index 'DC' INDEX/VSAM
.SEGM. length : 29 bytes
.RECORD length: 34 bytes
.DSN : &INDEX..&ROOT.&FILE.D3
.DBD : PACDD1xx
.Size : 30 records per 1024-CI

.The Cross-References file (DX)

.Organization : DL/1 HISAM/VSAM database
.SEGM. length : 80 bytes
.RECORD length: 4,096 bytes
.DSN : &INDEX..&ROOT.&FILE.DX
.DBD : PACDDXxx
.Size : 48 records per 4096-CI

.The DSMS Journal file (DJ)

.Organization : DL/1 HDAM/OSAM database
.SEGM. length : 187 bytes
.RECORD length: 4,096 bytes
.DSN : &INDEX..&ROOT.&FILE.DJ
.DBD : PACDD1xx
.Size : 20 records per 4096-CI

.The on-line DAF work file (DF)

.Organization : DL/1 HIDAM/VSAM database
.SEGM. length : 100 min., 554 max.
.RECORD length: 4,096 bytes
.DSN : &INDEX..&ROOT.&FILE.DF
.DBD : PACDDFxx
.Size : depending on records' length

.The on-line DAF work file primary index (F1)

.Organization : INDEX/VSAM 'DF' index
.SEGM. length : 37 bytes
.RECORD length: 39 bytes
.DSN : &INDEX..&ROOT.&FILE.F1
.DBD : PACDF1xx
.Size : 26 records per 1024-CI

.The DSMS Database of S.P.A.'s (DZ)

.Organization : DL/1 HIDAM VSAM database
.SEGM. length : 8,992 bytes
.RECORD length: 9,000 bytes
.DSN : &INDEX..&ROOT.&FILE.DZ
.DBD : PACDDZxx
.Size : 1 record per 10240-CI

Three other sequential files form the backup. They are:

.The Backup file (BB)

.Organization : Sequential generation file
.Recfm : Variable
.Lrecl : 354
.Blksize : 6,376
.DSN : &INDEXQ..&ROOT.&ROOT.BB(n)

.The Archived Journal file (BJ)

.Organization : Sequential generation file
.Reclsize : 180
.DSN code : &INDEXQ..&ROOT.&ROOT.BJ(n)

.The Deactivated Archived file (BQ)

.Organization : Sequential
.Reclsize : 180
.DSN code : User-defined

2.8. THE LIBRARY OF ENTRY-POINT SOURCES

LIBRARY OF ENTRY-POINT SOURCES

. Size : About 72 blocks of 6,080
. Organization: PDS
. DCB : Recfm=FB;Lrecl=80;Blksize=6,080
. DSNNAME : &INDEX..&ROOT.&ROOT.SRC

This complementary library contains the user check sub-programs for the definitions of Changes, Events, Sites, Requests and Reports.

! Member	! Contents	!
! xxCUAM	! TP check on Change definition	!
! xxCUEV	! TP check on Event definition	!
! xxCUMQ	! TP check on Report definition	!
! xxCURQ	! TP check on Query definition	!
! xxCUSI	! TP check on Site definition	!
! PDSCAM	! Batch check on Change definition	!
! PDSCEV	! Batch check on Event definition	!
! PDSCMQ	! Batch check on Report definition	!
! PDSCRQ	! Batch check on Query definition	!
! PDSCSI	! Batch check on Site definition	!

It also contains the DAF tables Dictionary (DAFDIC).

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
ENVIRONMENT

PAGE 26

3

3. ENVIRONMENT

3.1. ORGANIZATION OF DL/1 DATABASES

DL/1 DATABASE ORGANIZATION

DSMS is its own Database Manager. It uses DL/1 only to store those records supporting the physical organization of the database.

Consequently:

- . it is not possible to access DSMS data directly via DL/1 utilities. You must use the tools designed for this purpose and supplied by IBM.
- . DSMS requires very few DL/1 resources. Particularly, all of the databases are built from a single root Segment. Thus, there are no dependent segments, and the DBRs have fixed lengths.
- . Secondary indexes and logical relationships are not used. Therefore, the PHYSICAL ACCESS to DL/1 ACCESS ratio is considerably less than generally found in common applications.

Three types of DL/1 organization are used: HDAM, HISAM, and HIDAM.

HDAM-OSAM DATABASES

The HDAM organization is the organization reserved for the Journal database (DJ).

The key to these files is a seven-byte numeric field, filled in ascending order without sequence breaks. This record number is a logical pointer that is internal to the system.

Each record is attached to an anchor point in a one-to-one relationship (Anchor Point DL/1) calculated by the randomization module CGIPACR1, which is supplied with the system (in the SY parameters file under the name 'RANDOM'. It must be compiled in the IMS RESLIB of the installation site.). The CGIPACR1 module is directly derived from the randomization module modulo-DFSHDC10; the only difference is correction of the CI number which is done in order to avoid addressing in the 'BIT MAPS'.

Placement in the physical file is illustrated in the following example (assuming that each VSAM CI contains 25 DBRs):

DSMS key	CI Number	AP Number
1	2	1
2	2	2
3	2	3
..
..
25	2	25
26	3	1
27	3	2
...
etc...		

Consequently,

- . Synonym chains never occur, thus no FREE SPACE or OVERFLOW AREA is to be provided for.

. The physical file is used according to the ascending RBAs. Therefore, the OSAM space can be allocated according to the real volume occupied by the database, and independently of the number of CIs which could be addressed by the randomization module.

. The description of the RMNAME macro is:
RMNAME=(CGIPACR1,X,Y)

Assuming:

-X = the number of ANCHOR POINTs per CI.

X is calculated according to the size of the DSMS record and the size of the CI:

$$X = \text{INT}((CI - 19) / (REC + 11))$$

with:

INT = integer function.

CI = size of the CI in bytes.

REC = size of the DSMS record in bytes.

-Y = number of CIs which could be addressed.

Keeping in mind that B1 is an incremental function of the DSMS key, and that this key is assigned in ascending sequence, the user may choose a high value for the number of B1's (for example 16,000,000) without affecting physical organization or performance, thus avoiding possible overflows.

The value must not exceed ((2 exponent 24)-1).

EXAMPLE:

Let's consider the journal database (DJ):

```
.Size of the DSMS record      : 187
.Size of the CI in use        :4,096
---> X=20
---> RMNAME=(CGIPACR1,20,16000000)
```

HISAM VSAM DATABASES

This is the organization reserved for the Documentation database (DE), the Cross-References file (DX) and the SPA database (DZ).

The physical DL/1 record contains one and only one complete DBR, i.e. one and only one DSMS record plus DL/1 control information. There is no DATASET OVERFLOW.

During loading, the size of the database is directly deduced from: the number of logical records, the size of the RECORD, and the FREE SPACE requested at the time of the DEFINE of the VSAM file.

HIDAM VSAM DATABASES

This is the organization reserved for the DAF work file (DF) whose primary index is built by the F1 database, and for the database (DA) whose primary index is built by the D1 database.

Each DBR of these two databases (DA and DF) contains only one root segment of variable length.

Same organization for the DC database, the VisualAge Pacbase elements, shipped with the VA Pac System.

3.2. SYSTEM SPACE REQUIREMENTS

SYSTEM SPACE REQUIREMENTS

```
+-----+
! The file sizes are given in tracks for a 3390 disk !
! !
+-----+
! ! ON-LINE.....: 70 tracks !
! SYSTEM ! BATCH.....: 55 tracks !
! LIBRARIES ! 'SY' PARAMETERS.....: 9 tracks !
! ! DBDLIB.....: 2 tracks !
! ! PSBLIB.....: 5 tracks !
+-----+
! ! !
! SYSTEM ! DE0.....: !
! FILES ! DE.....: 75 tracks !
+-----+
! ! Tracks (3390).....: 220 tracks !
! T O T A L ! Bytes .....: 12.466.080 b. !
! ! (220 X 56,664) !
+-----+
```

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
BATCH PROCEDURES

PAGE 32

4

4. BATCH PROCEDURES

4.1. INTRODUCTION

INTRODUCTION

Batch processing with DSMS is divided into various procedures. The following chapters describe each of these procedures that may be used and give details on their specific execution conditions.

For each procedure, there is:

- . A general presentation containing:
 - an introduction,
 - the execution condition(s),
 - the action(s) to be taken in case of abnormal execution,
- . The description of user input, processing, results, and possible recommendations on use.
- . A description of each step containing:
 - The files used (temporary and permanent),
 - The return codes that may be generated by each step.

4.2. CLASSIFICATION OF PROCEDURES

CLASSIFICATION OF PROCEDURES

There are various types of batch procedures.

DATABASE MANAGEMENT PROCEDURES:

- . Initialization of DSMS files (DINI)
- . Archiving of file update transactions (DARC)
- . Restoration of files using the backup and archived files (DRST)
- . Backup of files (DSAV)
- . Reorganization of cross-references files (DREO).

UTILITY PROCEDURES:

- . Extraction, from the VA Pac Journal, of transactions corresponding to modified VA Pac entities that relate to changes (DEXP).
- . Extraction, from the DSMS journal (DXBJ), of transactions for update by the DUPT batch procedure.
- . Printing of query results, and of table and keyword lists requests (DPRT).
- . Extraction from DSMS of Events, Changes, Sites or Tables as batch transactions (DEXT).
- . Extraction of DSMS tables to create lists of external values for the revamped version of the developer's workstation (DEXH).
- . Batch update of DSMS files of Events, Changes, Sites or Tables (DUPT, DUPD).
- . Pre-processing of DAF source files (DPDF).
- . Renaming of Table, Site and Keyword codes (DREN).
- . Printing of the installed programs list (DINS).

RETRIEVAL OF PREVIOUS RELEASES

For sites where DSMS monitors control VA Pac Databases, the installation of DSMS 2.5 requires version 8.0.2, or higher, of VA Pac.

PREVIOUS RELEASE RETRIEVAL PROCEDURES:

- . Retrieval of DSMS 8.0.1 Database (DR80).
- . Retrieval of DSMS 8.0.2 Database compatible with VA Pac 8.0.1 (DR8X) to be used when switching over from VA Pac 8.0.1 to 8.0.2.
- . Retrieval of DSMS 8.0.2 v01 or v02 Database (DR8Q) (retrieval of queries).
- . Retrieval of DSMS 1.2 or 1.5 Database (DR15)
- . Retrieval of DSMS 1.5 archived journal (DR5J)

RETRIEVAL OF A DATABASE FOR ANOTHER PLATFORM:

- . Replacement of low-values by blanks (DLVB).

4.3. ABNORMAL EXECUTION

ABNORMAL EXECUTIONS

Batch programs may sometimes terminate abnormally. For example, input-output errors on the system files or on the database provoke an abnormal end with an ABEND USER (code 12) accompanied by a message on the SYSOUT file.

When an ABEND occurs, the user must find this error message. It is displayed in the following manner:

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

In most cases, an examination of the return code and of the type of operation will allow the user to find the cause of the abnormal end (unavailable resources, file too small, etc.).

If this message is absent and the type of ABEND generated directly signals a problem in the VA Pac System programs, contact the VA Pac Technical Support. Be sure to KEEP ALL LISTINGS that may be necessary to analyze the problem.

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
JOURNAL ARCHIVING

(DARC)

PAGE 37

5

5. JOURNAL ARCHIVING

(DARC)

5.1. INTRODUCTION

DARC: INTRODUCTION

The Journal Archiving procedure (DARC) backs up the Journal file (DJ) as a sequential file (BJ), and reinitializes it both logically and physically.

The new archived transactions do not overwrite transactions previously archived; they are added to them.

The previously archived transactions can be deactivated, if requested.

EXECUTION CONDITION

The database must be closed to on-line use.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

If the abnormal end occurs before the step which creates the Journal file (DJ), the procedure can be restarted as it is, after the problem has been solved.

If the abnormal end occurs during or after this step, the user input must be modified before a new execution of the procedure so as to specify a reinitialization request without a backup of the Journal file (already backed-up).

CAUTION

With systems using generation files (MVS for instance), the +1 version of the archived transaction file might have been cataloged even if the procedure ended abnormally. In this case, the procedure must be executed again with the -1 version of the archived transaction file (not the 0 version) as input.

5.2. INPUT - PROCESSING - RESULTS

USER INPUT

The DARC procedure includes an optional input for:

- . deactivating previously archived transactions that are now obsolete,
- . indicating the absence of previously archived transactions during input,
- . indicating the unavailability of the Data file (DA) during input,
- . requesting a reinitialization of the transaction file only.

The structure of this input is as follows:

!POS.!	LEN.!	VALUE	MEANING
! 2 !	! 1 !	! 'S' !	! Line code !
! 3 !	! 4 !	! nnnn !	! Session number !
! 7 !	! 8 !	! CCYYMMDD !	! OR date up to which the user requests !
! !	! !	! !	! deactivation !
! 15 !	! 1 !	! 'I' !	! Absence of previously archived !
! !	! !	! !	! transactions !
! 16 !	! 1 !	! 'D' !	! Data file (DA) unavailable !
! 17 !	! 1 !	! 'J' !	! Re-initialization without backup !

The session number and the date are exclusive. They are ignored if it is indicated that there are no previously archived transactions.

The unavailability of the Data file is to be indicated only when this file has been physically deleted (see paragraph 'RECOMMENDATIONS').

The reinitialization request without a backup is necessary when the Journal file is physically destroyed.

CAUTION:

In this case, the previous archiving is not duplicated on the output archiving. When the cataloging is automatic, previous archiving may be lost if no uncataloging is performed.

In case of an error on one of the options, an error message is printed and the archiving is generated using the default options.

RECOMMENDATIONS

If there is no user input, this procedure can be executed only if the database is in a consistent state, and if the Journal file is correctly formatted.

When a database needs to be restored after a problem, some information in the database may be destroyed and neither the DARC nor the DRST procedures can then be executed.

In this case, AND IN THIS CASE ONLY, columns 15 to 17 of the user input must be used as follows:

- . If the Data file (DA) is lost or has been flagged as 'inconsistent', a 'D' in column 16 means that the DARC procedure will not take the Data file (DA) into account. However, the DRST procedure must be executed afterwards, since under these conditions, the DARC procedure renders the database inconsistent.
- . If the Journal file (DJ) is lost or destroyed, a 'J' must be entered in column 17. The DARC procedure formats an empty Journal file. The DRST procedure can then be executed.
- . If the sequential Archived file (BJ) is lost or destroyed, an 'I' must be entered in column 15. The DARC procedure will format a new sequential archive file.

If one of these columns is accidentally set to its value, and the DARC procedure executed when the Data (DA) file is in a consistent state, the consequences are :

- . 'I' in col. 15: The transactions previously archived are lost. All the transactions can be recovered by concatenating BJ(-1) and BJ(0) to obtain BJ(+1).
- . 'D' in col. 16: The DARC procedure has to be re-run BEFORE any update. If it is done afterwards, the data is lost and a complete restoration must be executed.
- . 'J' in col. 17 : The contents of the Journal file are lost and cannot be retrieved.

REPORT RESULTS

This procedure prints a report giving the number of archived update transactions and, if applicable, the number of records that have been deactivated.

GENERAL RESULTS

Once this procedure is executed, a sequential file containing all the archived transactions is produced.

The Journal file is re-initialized.

It is also possible to store in another file all update transactions that have been deactivated.

NOTE: This procedure does not increment the current session number of the database.

5.3. DESCRIPTION OF STEPS

DARC: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

ARCHIVAL OF JOURNAL FILE: PDS300

This step executes the following:

- . Updates the file of archived update transactions,
- . Positions a flag in the Data file indicating the journal archiving,
- . Writes the deactivated transactions onto a special file, if deactivation is requested by user input.

.Input files:

- User transaction
PACDMB : DSN=&&PAC7MB
- Already archived transactions
PACDBJ: DSN=&INDEXQ.&ROOT.&FILE.BJ(0)
- Journal file to re-initialize :
PAC7DJ\$SUF: DSN=&INDEX.&ROOT.&FILE.DJ
- Error message file :
PAC7DE\$SUF: DSN=&INDEX.&ROOT.&ROOT.DE

.Input-Output file:

- Data file :
PAC7DA\$SUF: DSN=&INDEX.&ROOT.&FILE.DA
- PAC7D1\$SUF: DSN=&INDEX.&ROOT.&FILE.D1

.Output files:

- Archived update transactions
PACDBJ: DSN=&INDEXQ.&ROOT.&FILE.BJ(+1)
- Deactivated archived trans.
PACDBQ : DSN=DUMMY

(This file can be retrieved if necessary)

.Sort files:

SORTWK01, SORTWK02, SORTWK03

.Output report:

- Review of archival
PACDRU

.Return codes:

- 0: No error detected on the files
- 8: User Input error
- 12: Input-output error on a file.

DEFINITION OF THE JOURNAL FILE: IEFBR14

This step executes a DEFINE on the Journal file (DJ):

.Defined file:

-Journal file
PAC7DJ\$\$SUF DSN=&INDEX..&ROOT.&FILE.DJ

RE-INITIALIZATION OF THE JOURNAL FILE: PDS320

This step executes the following:

- . Creates a record in the Journal file
- . Repositions the Data file flag.

.Input files:

-User transaction
PACDMB : DSN=&&PAC7MB
-Error-message file
PAC7DE\$\$SUF: DSN=&INDEX..&ROOT.&ROOT.DE

.Input-Output file:

-Data file
PAC7DA\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D1

.Output file:

-Journal file to be re-initialized
PAC7DJ\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DJ

.Output report:

-Reinitialization report
PACDRU

5.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - ARCHIVAL OF DSMS JOURNAL -
//*****
//$RADP.DARC PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, DSMS SYSTEM ROOT
// INDEX='$INDEX', VSAM INDEX
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// VOLS='SER=$VOLO', VOLUME OF ARCHIVED TRANSACTIONS
// VOLU='SER=$VOLU', VOLUME OF DATABASE 'DJ'
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
// INDEXQ='$INDEXQ', INDEX OF DATA GROUP FILES
// SPABJ='TRK,(020,10)', SPACE FOR TRANSACTION FILE
// SPADJ='(CYL,(3,2))', SPACE FOR 'DJ' (OSAM DATABASE)
// UNITS='$UNITO', TRANSACTION FILE UNIT
// UNITU='$UNITU', UNIT OF DATABASE 'DJ'
// CYL=3, SORT WORKS SIZE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT', SORT LIBRARY
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* INPUT : COMMAND TO DEACTIVATE ARCHIVED TRANSACTIONS *
//* COL 2 : 'S' *
//* COL 3 - 6 : SESSION NUMBER *
//* COL 7 - 14 : DATE (CCYYMMDD) *
//* COL 15 : ' ' ----> PRESENCE OF ARCHIVED TRANSACTIONS FILE *
//* : 'I' ----> ABSENCE OF ARCHIVED TRANSACTIONS FILE *
//* COL 16 : ' ' ----> PRESENCE OF DATA DATABASE (DA) *
//* : 'D' ----> ABSENCE OF DATA DATABASE (DA) *
//* COL 17 : ' ' ----> ARCHIVING + REINITIALIZATION *
//* : 'J' ----> REINITIALIZATION WITHOUT ARCHIVING *
// *
// * IF THERE IS NO INPUT (OR IF THERE IS ANY ERROR IN IT), NO DEAC- *
// * TIVATION IS DONE, ARCHIVAL AND REINITIALIZATION ARE RUN NORMALLY *
// *
// * TRANSACTIONS WITH SESSION (DATE) LOWER OR EQUAL TO THE SESSION *
// * (DATE) INDICATED ARE NOT ARCHIVED. THEY ARE SAVED IN THE FILE *
// * CONTAINING THE DEACTIVATED TRANSACTIONS. *
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&PAC7MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
// *
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
// *
//PDS300 EXEC PGM=DFSRRC0,REGION=$REGSIZ,
// PARM=(DLI,PDS300,PDS300$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,

```

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

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

VisualAge Pacbase - Operations Manual	PAGE	47
DSMS - INSTALLATION & OPERATIONS		
PRINTING OF QUERIES AND OUTPUT REPORTS	(DPRT)	6

6. PRINTING OF QUERIES AND OUTPUT REPORTS (DPRT)

6.1. INTRODUCTION

DPRT: INTRODUCTION

The DPRT procedure performs all the printing operations for DSMS:

- . Results of User Queries on Events, Changes and Sites, (this order must be respected)
- . Printouts of Tables, Keywords, Queries and Reports.

See the DSMS Reference Manual for practical information on how to submit a DPRT execution in either batch or on-line mode.

NOTE: Printouts of Tables and Keywords can be submitted in batch mode only.

Technical information regarding the JOB Function allowing for DPRT on-line submissions is given at the end of this chapter.

EXECUTION CONDITIONS

None.

The Database can remain open to on-line processing.

ABNORMAL EXECUTION

Refer to Chapter THE BATCH PROCEDURES, Subchapter 'Abnormal Execution'.

6.2. INPUT - PROCESSING - RESULTS

USER INPUT

A '*' line (required):

```

+-----+-----+-----+-----+
!Col.! Len.! Value  ! Description  !
+-----+-----+-----+-----+
!  2 !   1 ! '*'    ! Line Code    !
!  3 !   8 ! uuuuuuu ! DSMS User Code  !
! 11 !   8 ! ppppppp ! Password      !
! 19 !   3 ! ppp     ! Product Code  !
! 22 !   2 ! su      ! Subsidiary Code !
! 24 !   1 ! 1       ! Language Code  !
+-----+-----+-----+-----+

```

4 report types exist, 1 line per printout is necessary :

```

+-----+-----+-----+-----+
!Col.! Len.! Value  ! Description  !
+-----+-----+-----+-----+
! TABLES
+-----+-----+-----+-----+
! 02 !  03 ! Txx    ! Table codes for Txx
! 06 !  02 ! C1     ! ... with their label in connected
!   !   !       ! user language (default option)
! 06 !  02 ! C2     ! ... with all labels
! 02 !  03 ! TUD    ! User codes with all authorizations
!   !   !       ! (TUG + TUP + TUS)
+-----+-----+-----+-----+
! QUERIES / REPORTS
+-----+-----+-----+-----+
! 02 !  04 ! X QC   ! Query on Changes
!   !   ! X QE   ! Query on Events
!   !   ! X QS   ! Query on Sites
! 02 !  04 ! X RC   ! Report on Changes
!   !   ! X RE   ! Report on Events
!   !   ! X RS   ! Report on Sites
! 06 !  06 ! xxxxxx ! Query or Report code
! 12 !  08 ! uuuuuuu ! User code for Query or Report owner
!   !   !       ! (default value: connected user code)
! 20 !  02 ! C1     ! Print of all description pages
!   !   !       ! for the Query/Report type
!   !   !       ! (default option)
!   !   ! C2     ! Print of only useful Query/Report
!   !   !       ! description lines
+-----+-----+-----+-----+

```

```

+-----+
!Col.! Len.! Value ! Description !
+-----+
! LISTS !
+-----+
! 02 ! 03 ! LJQ ! Control cards !
! 02 ! 04 ! LCQC ! Query on Changes !
! ! ! LCQE ! Query on Events !
! ! ! LCQS ! Query on Sites !
! 02 ! 04 ! LCRC ! Reports on Changes !
! ! ! LCRE ! Reports on Events !
! ! ! LCRS ! Reports on Sites !
! 07 ! 02 ! C1 ! Print of all description pages !
! ! ! ! for the Query/Report type !
! ! ! ! (default option) !
! ! ! C2 ! Print of only useful Query/Report !
! ! ! ! description lines !
! 12 ! 08 ! uuuuuuu ! User code for Query/Report owner !
+-----+
! KEYWORDS !
+-----+
! 02 ! 04 ! LAKC ! Stand-alone Keywords for Changes !
! ! ! LPKC ! Principal keywords for Changes !
! ! ! LGKC ! All keywords for Changes !
! 06 ! 01 ! 1 ! Keywords language code (default: !
! ! ! ! connected user language code) !
! 02 ! 04 ! LAKE ! Stand-alone Native Keywords for Evnts!
! ! ! LPKE ! Principal Native Keywords for Events !
! ! ! LGKE ! All Native Keywords for Events !
! 02 ! 04 ! LAKT ! Stand-alone Techn. Keywords for Evnts!
! ! ! LPKT ! All main keywords for Events !
! ! ! LGKT ! All keywords !
+-----+

```

```
+-----+
!Col.! Len.! Value  ! Description  !
+-----+
!   !   !   ! .PRINT VIA USER QUERY:  !
!  5 !  6 ! rrrrrr ! Code of the user Query (required)  !
!   !   !   ! 'Q' Entity used.  !
!  5 !  6 ! mmmmmmm ! Code of the Report (optional)  !
! 17 !  1 ! d      ! Delimiter      (optional)  !
!   !   !   ! Parameters:  !
! 18 !  1 ! s      ! Symbol          -  !
! 19 !  1 ! x      ! Separator       -  !
! 20 ! 54 ! ..... ! Parameter values -  !
!   !   !   ! If optional fields have not been  !
!   !   !   ! filled in, default values are used.  !
!   !   !   ! They come from the definition lines  !
!   !   !   ! of the user Query found in the Data-  !
!   !   !   ! Base.  !
+-----+
```

PRINTED OUTPUT

Two types of printed output are obtained:

- Results of user-defined Queries on Events, Changes and Sites.
- Standard printouts of Tables, Keywords, Queries and Reports.

RETURN CODE

```
+-----+
!  0 ! OK with Queries  !
!  4 ! OK with tables, kws, Queries/Reports print requests!
!  8 ! OK with erroneous Queries or other requests  !
! 12 ! Fatal error  !
! 16 ! Sort error  !
+-----+
```

6.3. DESCRIPTION OF STEPS

DPRT: DESCRIPTION OF STEPS

This procedure calls a unique program (PDSB) that acts as a flow monitor for the various programs, which are therefore sub-programs of this monitor.

The procedure includes the following steps:

INPUT RECOGNITION: PTU001

The input file is automatically formatted when QUERIES are submitted on-line.

VERIFICATION OF VSAM FILES: IDCAMS

PRINTING: PDSB

.Permanent input files:

- Data file
 - PAC7DA\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DA
 - PAC7D1\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D1
- VA Pac element file
 - PAC7DC\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DC
 - PAC7D3\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D3
- Error message file
 - PAC7DE\$\$SUF: DSN=&INDEX..&ROOT.&ROOT.DE

.Input file:

- User Queries
 - PACDMB : DSN=&&PACDMB

.Work files:

- Print requests
 - PACDKD
- Queries
 - PACDKQ
- Temporary files
 - PACDW1, PACDW2, PACDW3, PACDW4
- Sort files
 - SORTWK01, 02, 03

.Output reports:

- Flow report
 - PACDIA
- List of Queries and requests
 - PACDIB
- Print of tables and keywords
 - PACDID
- Report of Query extractions
 - PACDIQ
- Print of Query extractions
 - PACDQI
- Print of Queries/Reports
 - PACDRQ
- Print of control cards
 - PACDJQ

6.4. EXECUTION JCL

```

//*****
// * DSMS 2.5 *
// * - PRINT AND QUERY - *
//*****
//$RADP.DPRT PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, DSMS SYSTEM ROOT
// INDEX='$INDEX', INDEX OF VSAM FILES
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// OUTL='$OUT', OUTPUT CLASS OF REPORTS
// SPAMB='TRK,(100,10)', SPACE OF EXTRACTION COMMANDS
// SPAWK='CYL,(020,02)', SPACE WORK FILES
// UWK=$UWK, WORK UNIT
// CYL=3, SORT WORKS SIZE
// COPIES=1, NUMBER OF REPORT COPIES
// LNG='E', LANGUAGE OF MONITOR (ENGLISH 'E')
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BLIB', SORT LIBRARY
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// DBDLIO='$DBDLIO', LIBRARY OF VA PAC DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
// *-----*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPAMB),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSB EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSB&LNG,PDSB$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)

```

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

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
DATABASE RESTORATION

(DRST)

PAGE 55

7

7. DATABASE RESTORATION

(DRST)

7.1. INTRODUCTION

DRST: INTRODUCTION

The Database Restoration procedure (DRST) restores the files, using the sequential image produced by the Database Backup procedure (DSAV).

Archived transactions can also be retrieved once this procedure has been executed.

EXECUTION CONDITIONS

The database must be closed to on-line processing.

These modifications must be made in the System Parameters library.

The procedure physically and logically re-initializes the Journal file which must have been saved previously by the DARC procedure.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

Whatever the cause of the abend, the procedure can be restarted as it is, after the problem has been solved.

NOTE:

Since the DRST procedure recreates the files, it may be useful to adjust their size according to their estimated evolution.

DEFINITION CONTROL SUB-PROGRAMS

Sub-programs (delivered as COBOL sources) are designed to add specific controls or initializations on the 5 DSMS definitions.

At the beginning, these sources only include 3 examples:

- 1 'WARNING'-type error
- 1 critical error
- 1 initialization.

Their linkage is made up of the displayed fields, the entered fields or some other fields directly or indirectly associated with the definition.

At these sub-programs' return, an error message can then be displayed or the values of the displayed fields can be overwritten.

NOTES:

- . The usual controls on definitions are executed before and after their call.
- . When WARNING errors are set, a message is sent to the Definition screen and the sub-program is recalled to reinitialize the PR which is set to 'W'.

These sub-programs are called via tops indicated in the technical record of the DRST procedure.

7.2. INPUT - PROCESSING - RESULTS

USER INPUT

The following chart lists the DRST procedure's input.

!POS.!	!LEN.!	!VALUE	!MEANING
! 2 !	! 1 !	! 'R' !	! Line code !
! 3 !	! 1 !	! '1' !	! Language code 'E' or 'F' (optional) !
! 4 !	! 1 !	! !	! Journal inhibition flag !
! !	! !	! '0' !	! No inhibition (default option) !
! !	! !	! '1' !	! Inhibition !
! 5 !	! 1 !	! !	! Not used !
! 6 !	! 3 !	! 'REC' !	! Restoration and retrieval of archived !
! !	! !	! !	! transactions !
! 9 !	! 12 !	! !	! 12-position table indicating the !
! !	! !	! !	! PFkeys assignment !
! !	! !	! !	! (default: 123456789ABC, but you may !
! !	! !	! !	! move or set to blank one or several !
! !	! !	! !	! values) !
! 21 !	! 1 !	! !	! SECURITY SYSTEM INTERFACE !
! !	! !	! ' ' !	! Retrieval of the previous value or !
! !	! !	! !	! no interface (for creation) !
! !	! !	! '&' !	! Clear = Deactivation !
! !	! !	! 'R' !	! RACF !
! !	! !	! 'S' !	! TOPSECRET !
! 22 !	! 1 !	! !	! USER CONTROL USING ON-LINE RACF !
! !	! !	! ' ' !	! Retrieval of the previous value !
! !	! !	! '&' !	! Clear = it is possible to enter !
! !	! !	! !	! a user-password different from the !
! !	! !	! !	! one entered at the first connection !
! !	! !	! 'N' !	! It is not possible to enter another !
! !	! !	! !	! user-password !
! 23 !	! 1 !	! 'C' !	! Encryption of passwords !
! !	! !	! 'D' !	! Decryption of passwords !
! !	! !	! ' ' !	! Unchanged passwords !
! !	! !	! !	! NOTE: it is not advised at all to !
! !	! !	! !	! request an encryption or decryption !
! !	! !	! !	! of passwords at the same time as the !
! !	! !	! !	! retrieval of archived transactions !
! !	! !	! !	! request (because the action is not !
! !	! !	! !	! performed on the journal). !

```

+-----+-----+-----+-----+
!COL.! Len.! Value  ! Designation      !
+-----+-----+-----+-----+
! 26 !   1 ! 'C'  ! Call of the sub-routine of additional!
!   !   !     ! controls for Change definition      !
!   !   ! '&'  ! No call of sub-routine              !
! 27 !   1 ! 'E'  ! Call of the sub-routine of additional!
!   !   !     ! controls for Event definition       !
!   !   ! '&'  ! No call of sub-routine              !
! 28 !   1 ! 'Q'  ! Call of the sub-routine of additional!
!   !   !     ! controls for Query definition       !
!   !   ! '&'  ! No call of sub-routine              !
! 29 !   1 ! 'R'  ! Call of the sub-routine of additional!
!   !   !     ! controls for Report definition      !
!   !   ! '&'  ! No call of sub-routine              !
! 30 !   1 ! 'S'  ! Call of the sub-routine of additional!
!   !   !     ! controls for Site definition        !
!   !   ! '&'  ! No call of sub-routine              !
+-----+-----+-----+-----+

```

OUTPUT REPORT

This procedure prints a report listing the requested options, associated errors, the number of records restored in the database for each file, and the options memorized in the new database.

RESULT

Once this procedure is executed, the current session number is that of the sequential image or that of the most recent transaction, if the retrieval of archived transactions has been requested.

7.3. DESCRIPTION OF STEPS

DRST: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

Check on Journal existence: IDCAMS

This step executes a LISTCAT on the Journal file (DJ).
Its return codes are:

.0 : The journal file exists
.AUTRE : The journal file does not exist

VALIDATION OF JOURNAL CONTENTS: PDS380

This step is executed only when the Journal file exists. In
this case, it verifies that the journal has been archived.

.Input files:

-Journal file
PAC7DJ\$SUF: DSN=&INDEX..&ROOT.\$FILE.DJ
-Error message file
PAC7DE\$SUF: DSN=&INDEX..&ROOT.&ROOT.DE

.Output report:

-AJ file status report
PACDRU
It is printed if the journal file has not been archived.

.Return codes:

-0: The Journal file was archived.
-4: The Journal file was not archived.
(In this case, none of the DRST steps is executed).

DEFINITION OF FILES: IDCAMS

This step is executed only when the Journal file has been
archived. It executes a DELETE/DEFINE on the files of the
database.

DEFINITION OF FILE: IEFBR14

This step executes a DELETE/DEFINE on the Journal file.

DATABASE RESTORATION: PDS400

This step is executed only when the Journal file has been archived.

.Permanent input files:

-Backup of the files
PACDBB: DSN=&INDEXQ..&ROOT.&FILE.BB(0)
-Error message file
PAC7DE\$\$SUF DSN=&INDEX..&ROOT.&ROOT.DE

.Permanent output files:

-Data file
PAC7DA\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D1
-VA Pac element file
PAC7DC\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DC
PAC7D3\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D3
-Journal file
PAC7DJ\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DJ
-Cross-reference file
PAC7DX\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DX

.Input transaction file:

-User transactions
PACDMB : DSN=&&RESTMB

.Output file:

-Work file (2 records)
PACDMS : DSN=&&PACDMS

.Output report:

-Restoration report
PACDRU

RETRIEVAL OF ARCHIVED JOURNAL: PDS450

This step is executed only when there are transactions to be retrieved. It does not cause a 'journalization' of processed transactions.

.Permanent input-output files:

-Data file
PAC7DA\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D1
-VA Pac-element file
PAC7DC\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DC
PAC7D3\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D3
-Cross reference file
PAC7DX\$\$SUF: DSN=&INDEX..&ROOT.&FILE.DX

.Input files:

-Work file (2 records)
PACDMS : DSN=&&PACDMS
-Error message file
PAC7DE\$\$SUF: DSN=&INDEX..&ROOT.&ROOT.DE

.Input archived file:

-Archiving of the journal to retrieve
PACDBJ : DSN=&INDEXQ..&ROOT.&FILE.BJ(0)

.Output report:

-Update report
PACDRU

7.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - LOADING-RESTORATION OF DSMS DATABASE -
//*****
//$RADP.DRST PROC FILE=$FILE, PHYSICAL DATABASE NUMBER
// FILEO=$FILEO, VA PAC PHYSICAL DATABASE NUMBER
// ROOT=$ROOT, DSMS SYSTEM ROOT
// ROOTO=$ROOTO, VA PAC SYSTEM ROOT
// INDEX='$INDEX', VSAM INDEX
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// INDE XO='$INDE XO', NON VSAM VA PAC FILE INDEX
// INDE XP='$INDE XP', NON VSAM FILE INDEX
// INDE XQ='$INDE XQ', DATA GROUP FILE INDEX
// SPADJ='(CYL,(3,2))', SPACE 'DJ' (OSAM DATABASE)
// VOLU='$VOLU', VOLUME OF DATABASE 'DJ'
// UNITU='$UNITU', UNIT OF DATABASE 'DJ'
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// DBDLIO='$DBDLIO', LIBRARY OF VA PAC DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* INPUT
//* COL 02 : 'R'
//* COL 03 : INITIAL LANGUAGE CODE (F=FRENCH, E=ENGLISH)
//* COL 04 : '1' --> INHIBITION OF TRANSACTION LOG
//* COL 05 : MACHINE DATE FORMAT (I --> POUR MM/DD/YY)
//* : (N --> POUR DD/MM/YY)
//* COL 06-08 : 'REC' FOR RETRIEVAL OF ARCHIVED TRANSACTIONS
//* : 'INI' FOR INITIALIZATION OF THE FILES
//* COL 09-20 : 12-POSITION TABLE INDICATING THE PFKEYS ASSIGNM
//* COL 21 : SECURITY SYSTEM INTERFACE
//* : ' ' --> RETRIEVAL OF THE PREVIOUS VALUE OR NO
//* : INTERFACE (FOR CREATION)
//* : '&' --> CLEAR = DEACTIVATION
//* : 'R' --> RACF
//* : 'S' --> TOPSECRET
//* COL 22 : USER CONTROL USING ON-LINE RACF
//* : ' ' --> RETRIEVAL OF THE PREVIOUS VALUE
//* : '&' --> CLEAR = IT IS POSSIBLE TO ENTER A USER-
//* : PASSWORD DIFFERENT FROM THE ONE ENTERED AT
//* : THE FIRST CONNECTION
//* : 'N' --> IT IS NOT POSSIBLE TO ENTER ANOTHER USER
//* : PASSWORD
//* COL 23 : 'C' --> ENCRYPTION OF PASSWORDS
//* : 'D' --> DECRYPTION OF PASSWORDS
//* : ' ' --> UNCHANGED PASSWORDS
//* COL 26 : 'C' --> CALL OF THE SUB-ROUTINE OF ADDITIONAL
//* : CONTROLS FOR CHANGE DEFINITION
//* : '&' --> NO CALL OF SUB-ROUTINE
//* COL 27 : 'E' --> CALL OF THE SUB-ROUTINE OF ADDITIONAL
//* : CONTROLS FOR EVENT DEFINITION
//* : '&' --> NO CALL OF SUB-ROUTINE
//* COL 28 : 'Q' --> CALL OF THE SUB-ROUTINE OF ADDITIONAL
//* : CONTROLS FOR QUERY DEFINITION
//* : '&' --> NO CALL OF SUB-ROUTINE
//* COL 29 : 'R' --> CALL OF THE SUB-ROUTINE OF ADDITIONAL
//* : CONTROLS FOR REPORT DEFINITION
//* : '&' --> NO CALL OF SUB-ROUTINE
//* COL 30 : 'S' --> CALL OF THE SUB-ROUTINE OF ADDITIONAL
//* : CONTROLS FOR SITE DEFINITION
//* : '&' --> NO CALL OF SUB-ROUTINE

```

```

// *
// * IF THE JOURNAL DATABASE OF TRANSACTIONS ON DISK (DJ) IS NOT *
// * REINITIALIZED, THE RESTORE CHAIN IS NOT EXECUTED. *
// * IT IS THEREFORE NECESSARY TO EXECUTE THE 'DARC' PROCEDURE FIRST. *
// *****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&DRSTMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
// *
//EXISDJ EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(LI&ROOT.&FILE.DJ),DISP=SHR
// *
//PDS380 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDS380,PDS380$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
// COND=(00,NE,EXISDJ)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DJ$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,DISP=SHR
//PACDRU DD SYSOUT=&OUT
// *
//DEFINE EXEC PGM=IDCAMS,COND=(00,NE,PDS380)
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.DA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.D1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILEO.DC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILEO.D3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.DX),DISP=SHR
// *
//OSAMDJ1 EXEC PGM=IEHPRGM,COND=(00,NE,PDS380)
//DD1 DD UNIT=&UNITU,VOL=SER=&VOLU,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(DL&ROOT.&FILE.DJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
// *
//OSAMDJ2 EXEC PGM=IEFBR14,COND=(00,NE,PDS380)
//PAC7DJ$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,UNIT=&UNITU,
// DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
// DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
// SPACE=&SPADJ
// *
//PDS400 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDS400,PDS400$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,&IRLM),
// COND=(00,NE,PDS380)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR

```



```
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//          DD DSN=&DBDLIO,DISP=SHR
//*:STEPSCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC$$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3$$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DJ$$SUF DD DSN=&INDEX..&ROOT.&FILE.DJ,DISP=SHR
//PAC7DX$$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PACDBB   DD DSN=&INDEXQ..&ROOT.&FILE.BB(0),DISP=OLD
//PACDMB   DD DSN=&&DRSTMB,DISP=(OLD,PASS)
//PACDMS   DD DSN=&&PACDMS,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//          SPACE=(TRK,(1,1))
//PACDRU   DD SYSOUT=&OUT
//*
//PDS450   EXEC PGM=DFSRRRC00,REGION=$REGSIZ,
//          PARM=(DLI,PDS450,PDS450$$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//          COND=((00,NE,PDS380),(00,NE,PDS400))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//          DD DSN=&DBDLIO,DISP=SHR
//*:STEPSCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC$$SUF DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3$$SUF DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DX$$SUF DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PACDBJ   DD DSN=&INDEXQ..&ROOT.&FILE.BJ(0),DISP=SHR
//PACDMS   DD DSN=&&PACDMS,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//*
```

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
DATABASE BACKUP

(DSAV)

PAGE 66

8

8. DATABASE BACKUP

(DSAV)

8.1. INTRODUCTION

DSAV: INTRODUCTION

The purpose of the backup procedure (DSAV) is to convert the main files that make up DSMS into a BB sequential format.

The backed-up files are :

- . The Data file (DA),
- . The VA Pac Element file (DC),
- . The Cross-reference file (DX).

EXECUTION CONDITION

The database must be closed to on-line processing in order to ensure its consistency during the execution of the DSAV procedure.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

The main cause of an abend is that the database has not been closed to on-line use.

After correction, the procedure can be restarted as it is.

8.2. INPUT - PROCESSING - RESULTS

USER INPUT

One optional line code.

```
-----  
!Col.! Len.! Value  ! Designation  !  
!-----+-----+-----!  
!  2 !   1 ! 'O'   ! Line Code    !  
!  3 !   3 ! 'ENC' ! Encryption of passwords !  
!    !    ! 'DEC' ! Decryption of passwords !  
!    !    ! ' '   ! Unchanged passwords  !  
-----
```

REPORT RESULTS

Once the backup is executed, a report is printed. It includes the number of records saved in each file and the session number.

OUTPUT RESULT

The output is a single sequential file (BB) of variable length, containing the image of the three saved files.

If the database is in an inconsistent state as a result of an abnormal end in the last update, the DSAV procedure is not executed.

NOTE:

The DSAV procedure increments the current session number.

8.3. DESCRIPTION OF STEPS

DSAV: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE CONSISTENCY CHECK: PDSBAS

.Permanent input files:
-Data file
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D1
-Error message file
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

.Output report
-Validity report
PACDRS

Return code
-This utility sends a return code 4 and causes an ABEND
in case of database invalidity.

DATABASE BACKUP: PDS500

.Input-Output file:
-Data file
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

.Permanent Input files:
-VA Pac-element file
PAC7DC\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
PAC7D3\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
-Cross-reference file
PAC7DX\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DX
-Error message file
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

.Input transaction file:
-User transactions
PACDMB : DSN=&&DSAVMB

.Output file:
-Sequential image of files
PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

.Output report:
-Backup report
PACDRU

8.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - BACKUP OF THE DSMS DATABASE -
//*****
//$RADP.DSAV PROC FILE=$FILE, NUMBER OF PHYSICAL BASE
// ROOT=$ROOT, ROOT OF DSMS SYSTEM
// INDEX='$INDEX', VSAM INDEX
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// INDEXP='$INDEXP', INDEX OF NON VSAM FILES
// INDEXQ='$INDEXQ', DATA GROUP FILE INDEX
// VOLS='SER=$VOLO', VOLUME OF GENERATION FILE
// UNITS='$UNITO', UNIT OF GENERATION FILE
// SPABB='TRK,(10,2)', SPACE OF BACKUP
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// DBDLIO='$DBDLIO', LIBRARY OF VA PAC DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK='$UWK', WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&DSAVMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR
//*
//PDSBAS EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSBAS,PDSBAS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)

```

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

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
REORGANIZATION OF CROSS-REFERENCE FILE

(DREO)

PAGE 72

9

9. REORGANIZATION OF CROSS-REFERENCE FILE (DREO)

9.1. INTRODUCTION

INTRODUCTION

The Cross-Reference Reorganization procedure (DREO) rebuilds a sequential image of the database using another sequential image as a starting point. The resulting file is used as input to the Restoration (DRST) procedure.

The operating principle of this procedure is to rebuild the cross-references associated with the data from the 'image' of this data.

EXECUTION CONDITIONS

The database can remain open during reorganization since the procedure operates on sequential images of the database (backups).

The updates executed after the file backing up used for reorganization, can be retrieved during the restoration of the reorganized database.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

In case of an abnormal end, the procedure must be restarted from the beginning.

9.2. INPUT - PROCESSING - RESULTS

USER INPUT

Three different types of user input can be entered, but only one line of each type can be created.

The format of this input is given in the table below.

!POS.!	!LEN.!	!VALUE	!MEANING
! 1 !	! 1 !	!Not Used!	!
! 2 !	! 1 !	!'P'	! Deletion of Products
! !	! 1 !	!'S'	! Deletion of Subsidiaries
! !	! 1 !	!'X'	! Deletion of Product/Subsidiary
! 3 !	! 60 !	!Product	! (20 x 3 char.) if Col.2 = 'P'
! !	! !	! code	!
! !	! 60 !	!Subsid.	! (30 x 2 char.) if Col.2 = 'S'
! !	! !	! code	!
! !	! 60 !	!Prod./	! (12 x 5 char.) if Col.2 = 'X'
! !	! !	!Subsid.	!
! !	! !	!	!

REPORT

This procedure prints messages stating inconsistencies found in the Data file.

RESULT

The result of this procedure is a reorganized sequential image of the DSMS database, used as input to the Restoration (DRST) procedure.

9.3. DESCRIPTION OF STEPS

DREO : DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

BUILDING OF INDEXES (not keywords): PDSR10

.Input file:

-Input file

CARTE : DSN=&&PACDMB

.Permanent Input files:

-DSMS database backup

PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(0)

-Error message file

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

.Work files:

-Data and VA Pac elements

PACDW1 : DSN=&&W1

-Keywords and keyword references

PACDW2 : DSN=&&W2

-Cross-references (not keywords)

PACDW3 : DSN=&&W3

-Sort files

SORTWK01, 02, 03

.Output reports:

-Inconsistencies in DSMS data

PACDRH

-Reorganization report

PACDRK

BUILDING OF KEYWORD INDEXES: PDSR20

.Work files:

-Keywords and keyword references

PACDW2 : DSN=&&W2

-Keywords

PACDW4 : DSN=&&W4

-Keyword references

PACDW5 : DSN=&&W5

-Sort files

SORTWK01, 02, 03

MERGE OF INDEXES: PDSR30

.Work files:
-Cross-references (except keywords)
PACDW3 : DSN=&&W3
-Keyword references
PACDW5 : DSN=&&W5
-Keyword references
PACDW6 : DSN=&&W6
-Sort files
SORTWK01, 02, 03

GENERAL MERGE FOR BACKUP: PDSR40

.Work files:
-Data and VA Pac elements
PACDW1 : DSN=&&W1
-Keywords
PACDW4 : DSN=&&W4
-Keyword references
PACDW6 : DSN=&&W6
-Sort files
SORTWK01, 02, 03

.Permanent input file:
-Error message file
PAC7DE\$SUF : DSN=&INDEXQ.&ROOT.&ROOT.DE

.Permanent output file:
-Reorganized DSMS database backup
PACDBB : DSN=&INDEXQ.&ROOT.&FILE.BB(+1)

.Output report:
-Reorganization report
PACDRR

9.4. EXECUTION JCL

```

//*****
// * DSMS 2.5
// * - DSMS REORGANIZATION - *
//*****
//$RADP.DREO PROC FILE=$FILE, PHYSICAL DATABASE NUMBER
// ROOT=$ROOT, DSMS SYSTEM ROOT
// OUT='$OUT', OUTPUT CLASS
// INDEX='$INDEX', VSAM INDEX
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
// INDEXQ='$INDEXQ', DATA GROUP FILE INDEX
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
// CYL=(4,1), SORT WORKS SIZE
// SPADA='TRK,(60,5)', WORK SPACE (DA + DC)
// SPADX='TRK,(60,5)', WORK SPACE (DX)
// SPABB='TRK,(20,5)', SPACE OF BACKUP
// VOLS='SER=$VOLO', VOLUME OF GENERATION FILE
// UNITS='$UNITO', UNIT OF GENERATION FILE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT', SORT LIBRARY
// UWK=$UWK, WORK UNIT
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----*
// *
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
// *
//PDSR10 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSR10,PDSR10$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP.&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DE$SUF DD DSN=&INDEX.&ROOT.&ROOT.DE,DISP=SHR
//CARTE DD DSN=&&PACDMB,DISP=(OLD,DELETE)
//PACDBB DD DSN=$INDEXQ.&ROOT.&FILE.BB(0),DISP=SHR
//PACDW1 DD DSN=&&W1,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPADA,RLSE),
// DCB=(RECFM=VB,LRECL=354,BLKSIZE=6022)
//PACDW2 DD DSN=&&W2,DISP=(,PASS),

```

```

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

```

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

VisualAge Pacbase - Operations Manual	PAGE	80
DSMS - INSTALLATION & OPERATIONS		
EXTRACTION FROM VA PAC ARCHIVED JOURNAL (DEXP)		10

10. EXTRACTION FROM VA PAC ARCHIVED JOURNAL (DEXP)

10.1. INTRODUCTION

EXTRACTION FROM ARCHIVED JOURNAL (DEXP): INTRODUCTION

The Archived Journal Extraction procedure (DEXP) extracts transactions associated to Changes from the VA Pac Archived Journal file, and formats them in order to update, in the DSMS Database, the modified elements corresponding to each Change.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

If an abnormal end occurs, the procedure can be restarted as it is, after the problem has been solved.

NOTES:

The DEXP procedure operates with a VA Pac 2.0 or higher Journal.

The DEXQ procedure operates with a Journal in a VA Pac release lower than 2.0.

10.2. INPUT - PROCESSING - RESULTS

USER INPUT

One '*'-line is required:

! POS.!	! LEN.!	! VALUE	! MEANING	!
! 2	! 1	! '*'	! Line code	!
! 3	! 8	! uuuuuuuu!	! DSMS user code	!
! 11	! 8	! pppppppp!	! User password	!

One extraction line is also required:

! POS.!	! LEN.!	! VALUE	! MEANING	!
! 2	! 1	! 'J'	! Line code (required)	!
!	!	!	! THE FOLLOWING FIELDS ARE OPTIONAL :	!
! 3	! 1	! ' '	! List of selected transactions	!
!	!	! 'N'	! No list	!
! 4	! 24	!	! Selection in the VA Pac Database:	!
! 4	! 4	! nnnn	! Session number, begin. of selection	!
! 8	! 4	! pppp	! Session number, end of selection	!
!	!	!	! --> Selection on session(s)	!
!	!	!	! prohibits selection on date(s)	!
! 12	! 8	! CCYYMMDD!	! Starting date for selection	!
!	!	! 'TODAY'	! Starting date = current date	!
! 20	! 8	! CCYYMMDD!	! Ending date for selection	!
!	!	! 'TODAY'	! Ending date = current date	!
!	!	!	! (default value if st. date ='today')	!
! 28	! 1	!	! Version of selected transactions	!
!	!	! ' '	! Selection of all sessions	!
!	!	! 'T'	! Selection of frozen session	!
!	!	! 'Z'	! Selection of current session	!
! 29	! 3	! ppp	! Product code	!
! 32	! 4	! xxxx	! VA Pac Database logical code	!
! 36	! 3	! lll	! Code of selected library	!
! 39	! 16	!	! Type of selected entities	!
! 55	! 1	! ' '	! Extraction of transactions made	!
!	!	!	! under change 999999	!
!	!	! 'N'	! No extraction of 999999-change	!
!	!	!	! transactions	!
! 56	! 1	! ' '	! Printing of duplicate transactions	!
!	!	!	! for the same VA Pac entity	!
!	!	! 'N'	! No printing of duplicate transact-	!
!	!	!	! ions	!
! 57	! 6	! nnnnnn	! Change number	!

REPORT

Extraction report showing the list of formatted transactions.

RESULT

A DSMS database update transaction file to be used as input to the DUPT procedure.

10.3. DESCRIPTION OF STEPS

DEXP: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

TRANSACTION EXTRACTION AND FORMATTING: PDS600

.Permanent input files:

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

-Error message file

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

-VA Pac archived journal

PAC7PJ : DSN=&PAC7PJ

.Input transaction file:

-User transactions

PACDMB : DSN=&&EXPJMB

.Sort files:

.Output file:

-Update transaction file for DUPT

PACDMV : DSN=&&PACDMV

.Output report:

-Report on selection request

PACDRU

.Return codes:

- 0: No error and no list requested

-04: No error and printout of the transactions list

-08: Error on the user line code or parameter line

-12: I/O error on a file

PRINTING OF DSMS UPDATE TRANSACTIONS: PDS610

.Permanent input files:

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

-Error message file

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

.Input File:

-DSMS update transactions file

PACDMV : DSN=&&PACDMV

.Output report:

-List of update transactions

PACDRU

.Return codes:

- 0: No error

-12: I/O error on a file

10.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - EXTRACTION-UPDATE OF DSMS DATABASE -
//*****
//$RADP.DEXP PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, DSMS SYSTEM ROOT
// INDEX='$INDEX', VSAM INDEX
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// INDEXP='$INDEXP', INDEX OF NON VSAM FILES
// SPAMV='TRK,(2,1)', SPACE OF EXTRACTED TRANSACTIONS
// PAC7PJ='NULLFILE', DSN OF ARCHIVED TRANSACTIONS FILE
// CYL=3, SORT WORKS SIZE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BLIB', SORT LIBRARY
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* INPUT USER
//* COL 02 : *
//* COL 03 : USER CODE DSMS
//* COL 11 : PASSWORD
//* INPUT COMMAND LINE(S) FOR EXTRACTION
//* COL 02 : J
//* COL 03 : ' ' ----> SELECTED TRANSACTIONS LIST
//* : 'N' ----> NO LIST OF SELECTED TRANSACTIONS
//* COL 04-07 : STARTING SESSION NUMBER
//* COL 08-11 : ENDING SESSION NUMBER
//* COL 12-19 : STARTING DATE (CCYYMMDD)
//* COL 20-27 : ENDING DATE (CCYYMMDD)
//* COL 28 : VERSION OF SELECTED TRANSACTIONS
//* : ' ' ----> ALL SESSIONS
//* : 'T' ----> FROZEN SESSIONS
//* : 'Z' ----> CURRENT SESSION
//* COL 29-31 : PRODUCT CODE
//* COL 32-35 : INTERNAL VA PAC DATABASE CODE
//* COL 36-38 : LIBRARY CODE
//* COL 39-54 : TYPE OF ENTITIES TO BE SELECTED
//* COL 55 : ' ' ----> EXTRACTION OF TRANSACTIONS TRANSFERRED
//* : WITH CHANGE 999999
//* : 'N' ----> CHANGE 999999 TRANSACTIONS NOT EXTRACTED
//* COL 56 : ' ' ----> PRINTING OF DUPLICATE TRANSACTIONS ON
//* : A VA PAC ENTITY
//* : 'N' ----> DUPLICATE TRANSACTIONS NOT PRINTED
//* COL 57-62 : CHANGE NUMBER
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&EXPJMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX.&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX.&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX.&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
    
```

EXTRACTION FROM VA PAC ARCHIVED JOURNAL (DEXP)
EXECUTION JCL

10
4

```
//          DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDS600 EXEC PGM=DFSRRRC00,REGION=$REGSIZ,
//      PARM=(DLI,PDS600,PDS600$$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDMB   DD DSN=&&EXPJMB,DISP=(OLD,PASS)
//PACDMV   DD DSN=&&PACDMV,DISP=(,PASS),
//      UNIT=&UWK,SPACE=(&SPAMV,RLSE),
//      DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PAC7PJ   DD DSN=&PAC7PJ,DISP=SHR
//PACDRU   DD SYSOUT=&OUT
//*
//PDS610 EXEC PGM=DFSRRRC00,REGION=$REGSIZ,
//      PARM=(DLI,PDS610,PDS610$$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//      COND=(4,NE,PDS600)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDMV   DD DSN=&&PACDMV,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//*
```

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
EXTRACTION OF ENTITIES

(DEXT)

PAGE 87

11

11. EXTRACTION OF ENTITIES

(DEXT)

11.1. INTRODUCTION

ENTITY EXTRACTION (DEXT): INTRODUCTION

The Entity Extraction procedure (DEXT) extracts all DSMS entities and formats them into batch transactions to be used as input to the DSMS Database Update procedure (DUPT).

PRINCIPLE

In order to select the extraction of Changes, Events or Sites, the procedure uses Queries ("Q" entities) that must have been previously defined in the DSMS Database. These three types of extraction must be requested in the above order.

The Query code should also be specified in the extraction request (see 'User Input').

The screen Report ("R" entity) associated with the Query used for the extraction does not interfere in the extraction.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

If an abnormal end occurs, the procedure can be restarted as it is after the problem has been solved.

11.2. INPUT - PROCESSING - RESULTS

USER INPUT

One '*'-line is required:

```
+-----+-----+-----+-----+
! POS.! LEN.! VALUE ! MEANING !
+-----+-----+-----+-----+
!  2 !  1 ! '*'  ! Line code !
!  3 !  8 ! !uuuuuuu! DSMS User code !
! 11 !  8 ! !pppppppp! User password !
! 19 !  3 ! !ppp    ! Product code !
! 22 !  2 ! !su     ! Subsidiary code !
! 24 !  1 ! !l     ! Language code !
+-----+-----+-----+-----+
```

Four types of extractions are available. One line per request is necessary:

```
+-----+-----+-----+-----+
!Pos.! Len.! Value ! Meaning !
+-----+-----+-----+-----+
! 02 !  03 ! 'PL'  ! Locking of databases !
+-----+-----+-----+-----+
! 02 !  03 ! Txx   ! Codes of the Txx table !
!   !   !      ! (all tables except TRA) !
+-----+-----+-----+-----+
! QUERIES / REPORTS: !
+-----+-----+-----+-----+
! 02 !  04 ! X QC  ! Query on Changes !
!   !   ! X QE  ! Query on Events !
!   !   ! X QS  ! Query on Sites !
! 02 !  04 ! X RC  ! Report on Changes !
!   !   ! X RE  ! Report on Events !
!   !   ! X RS  ! Report on Sites !
! 12 !  08 ! !uuuuuuu! Owner of the Query or Report !
!   !   !      ! (Default=logged-in user) !
+-----+-----+-----+-----+
```

!Pos.!	Len.!	Value	!	Meaning	!
! 02 !	! 04 !	! LCQC	!	! Queries on Changes	!
!	!	! LCQE	!	! Queries on Events	!
!	!	! LCQS	!	! Queries on Sites	!
! 02 !	! 04 !	! LCRC	!	! Reports on Changes	!
!	!	! LCRE	!	! Reports on Events	!
!	!	! LCRS	!	! Reports on Sites	!
! 12 !	! 08 !	!uuuuuuuu!	!	! Owner of Queries or Reports	!
! KEYWORDS:					!
! 02 !	! 04 !	! LAKC	!	! Isolated keywords of Changes	!
!	!	! LGKC	!	! All Changes' Keywords	!
! 06 !	! 01 !	! 1	!	! Language code of Keywords	!
!	!	!	!	! (Default=Language of logged-in user)	!
! 02 !	! 04 !	! LAKE	!	! Native isolated Keywords of Events	!
!	!	! LGKE	!	! All Events' Keywords	!
! 02 !	! 04 !	! LAKT	!	! Techn. isolated Keywords of Events	!
!	!	! LGKT	!	! All Keywords	!

```
+-----+
!Pos.! Len.! Value  ! Meaning
+-----+
!      !      !      ! .EXTRACTION VIA USER QUERY:
!  5 !  6 ! rrrrrr ! User Query code (required)
!      !      !      ! - 'Q' Entity use
!  5 !  6 ! mmmmmm ! Report code (optional)
! 17 !  1 ! d      ! Delimiter (optional)
!      !      !      ! Parameter settings:
!      !      !      ! -----
! 18 !  1 ! s      ! Symbol -
! 19 !  1 ! x      ! Separator -
! 20 ! 54 ! ..... ! Parameter values -
!      !      !      !
!      !      !      ! If some optional fields were not
!      !      !      ! completed, default values will be
!      !      !      ! used. They come from the User
!      !      !      ! Query's definition lines found in
!      !      !      ! the Database.
+-----+
```

PRINTED OUTPUT

Extraction report showing the number of extracted transactions.

RESULT

DSMS database update transactions to be used as input to the DUPT procedure.

This procedure displays a general return code:

```
+-----+
!  0 ! OK
!  8 ! Error on the user line
!      ! code or on a command line
! 12 ! I/O error on a file or
!      ! Inconsistency of DSMS database
! 16 ! Sort error
+-----+
```

11.3. DESCRIPTION OF STEPS

DEXT: DESCRIPTION OF STEPS

This procedure calls a single program (PDSEX) that acts as a flow monitor for all programs, which are then considered as its sub-programs.

The procedure includes the following steps:

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

EXTRACTIONS: PDSEX

.Permanent input files:

-Data file
PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
-VA Pac element file
PAC7DC\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
PAC7D3\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
-Error message file
PAC7DE\$\$SUF : DSN=&INDEX..&ROOT.ROOT.DE

.Input transaction file:

-Extraction requests
PACDMB : DSN=&&PACDMB

.Work files:

-Queries
PACDKQ
-Temporary files
PACDW0, W1, W2, W3
PACDW4, W5, WI

.Output reports:

-Flow report
PACDIA
-Extraction request report
PACDRU

.Sort files:

SORTWK01, 02, 03

.Output file:

-Extracted batch transactions
PACDIM : DSN=&&PACDIM

11.4. EXECUTION JCL

```

//*****
// * DSMS 2.5 *
// * - EXTRACTION OF BATCH TRANSACTIONS FOR DUPT - *
//*****
//$RADP.DEXT PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, DSMS SYSTEM ROOT
// LNG='E', LANGUAGE OF MONITOR
// INDEX='$INDEX', VSAM INDEX
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// OUTL='$OUT', OUTPUT CLASS OF REPORTS
// SPAMB='TRK,(100,10)', SPACE OF EXTRACTION COMMANDS
// SPAIM='TRK,(100,10)', SPACE OF EXTRACTED TRANSACTIONS
// SPAWK='CYL,(020,02)', SPACE OF WORK FILES
// CYL=3, SORT WORKS SIZE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT', SORT LIBRARY
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// DBDLIO='$DBDLIO', LIBRARY OF VA PAC DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
// *-----*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX.&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX.&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX.&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX.&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX.&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP.&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSEX EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSEX&LNG,PDSEX$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)

```

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

VisualAge Pacbase - Operations Manual	PAGE	95
DSMS - INSTALLATION & OPERATIONS		
EXTRACTION OF TABLES FOR EXTERNAL LISTS (DEXH)		12

12. EXTRACTION OF TABLES FOR EXTERNAL LISTS (DEXH)

12.1. INTRODUCTION

DEXH: INTRODUCTION

The DEXH procedure extracts all the information contained in DSMS tables in order to create a file that can be used by a developer's workstation.

With the resulting file, the developer can create 'Lists of external values', used by the 'revamped' (using the PAW function) DSMS workstations.

For further details, see the PAW OPERATOR'S HANDBOOK, chapter 'REVAMPING OF IBM PRODUCTS'.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

Whatever the cause of the abend, the procedure can be restarted as it is, once the problem has been solved.

12.2. INPUT - PROCESSING - RESULTS

USER INPUT

```
+-----+
! Pos. ! Len. ! Value      ! Meaning
+-----+
!  2  !  1  ! '*'       ! Line code
!  3  !  8  ! uuuuuuuu  ! DSMS User code
! 11  !  8  ! pppppppp  ! Password
! 19  !  3  ! ppp       ! Product code
! 22  !  2  ! su        ! Subsidiary code
+-----+
```

REPORT

Extraction report showing the list of extracted tables.

RESULT

All general tables (not linked to a specific product) as well as the OPTIONS, PHASES and VERSIONS tables of the product specified in the user input.

12.3. DESCRIPTION OF STEPS

DEXH: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

EXTRACTION FOR PAW WORKSTATIONS: PDSXTH

This program extracts the values contained in tables: TST
TSU, TGR, TPR, TRE, TTY, TUD, TAT, TLA, TPH, and TOP
to be read on 'revamped' DSMS workstations.

.Permanent input files:

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

-Error message file

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

.Input transaction file:

-User check

PACDMB : DSN=&&PACDMB

.Output file:

-Extracted tables

PACDMV : DSN=&&PACDMV

.Output report:

-Extraction report

PACDRH

.Sort files:

SORTWK01, 02, 03

12.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - EXTRACTION OF TABLES FOR EXTERNAL LISTS -
//*****
//$RADP.DEXH PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, VA PAC-DSMS SYSTEM ROOT
// INDEX='$INDEX', VSAM INDEX
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
//* SYSTCAT='$SCATV', VSAM SYSTEM CATALOG
//* VSAMCAT='$SCATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// OUTL='$OUT', OUTPUT CLASS OF REPORTS
// SPAMV='TRK,(100,10)', SPACE OF EXTRACTED TRANSACTION
// CYL=3, SORT WORKS SIZE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT', SORT LIBRARY
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// DBDLIO='$DBDLIO', LIBRARY OF VA PAC DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//* DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSXTH EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSXTH,PDSXTH$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
// DD DSN=&DBDLIO,DISP=SHR
//*STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//* DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
    
```

EXTRACTION OF TABLES FOR EXTERNAL LISTS (DEXH)
EXECUTION JCL

PAGE

100

12

4

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

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
BATCH UPDATE OF ENTITIES

(DUPT)

PAGE 101

13

13. BATCH UPDATE OF ENTITIES

(DUPT)

13.1. INTRODUCTION

DUPT: INTRODUCTION

The Batch Update of Entities procedure (DUPT) updates the DSMS entities with transactions from the DEXT, DEXP and/or DXBJ procedures.

Transactions can also be entered directly in a file, using an editor. For a complete description of the batch transactions, see the 'BATCH TRANSACTIONS STRUCTURE', in the appendix of the DSMS Reference Manual.

EXECUTION CONDITION

The DSMS files must be closed to on-line use.

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

Whatever the cause of theabend, the procedure can be restarted as it is after the problem has been solved.

CAUTION:

This procedure performs a GLOBAL update. Therefore, make sure that all the data fields have been filled in. The data fields that are not filled in will automatically be set to blank.

The Change, Event and Site definition screens require two update lines, and both lines must be filled.

DSMS automatically allocates numbers to Events or Changes when they are created. However, for its creation, an Event or Change must be allocated a temporary number. For example, to create a Change: C000001, where 000001 is the temporary number that DSMS will automatically replace with a unique number.

You must set the action code to 'C', since the system does not provide for implicit creation.

Several Changes or Events can be created simultaneously. In this case, each Change or Event being created must be allocated a different temporary number. For example, to create 3 Changes simultaneously: C000001, C000002 and C000003.

NOTE: Each transaction stream can only contain 2,520 changes and 2,520 events maximum (internal limit of the program).

13.2. INPUT - PROCESSING - RESULTS

USER INPUT

- . One Parameter line (optional).
- . One Identification line per Product/Subsidiary concerned by the updates (required).
- . Update transactions extracted and formatted by the DEXT, DEXP or DXBJ procedures.
- . The user must add at least one identification line in front of update transactions.

Parameter line (optional)

```
+-----+
!Col Len! Value ! Description !
+-----+-----+-----+
! 2 1 ! $ ! LINE CODE !
! 3 1 ! ! UPDATE MODE / SORT ORDER !
! ! ! Defines the update or processing mode to !
! ! ! be used by ALL userids for this execution!
! ! ! the DSMS batch procedure. !
! ! A ! NORMAL UPDATE MODE !
! ! ! - Transactions sorted in ascending order !
! ! ! before any update is applied (i.e enti- !
! ! ! ty definitions are processed before !
! ! ! sub-screen records.) !
! ! ! - Update mode specified for each sign-on !
! ! ! record. !
! ! D ! DELETE MODE !
! ! ! - Transactions sorted in descending order!
! ! ! before any update is applied. !
! ! ! - All transactions processed as Deletions!
! ! ! - Action Code D'. !
! ! ! - Sign-on records must specify 'NORMAL' !
! ! ! mode - all other modes are considered !
! ! ! as errors. !
+-----+-----+-----+
```

```
+-----+
!Col Len! Value ! Description                                     !
+-----+-----+-----+
!  4  1 !          ! REPORT FORMAT INDICATOR                                     !
!          ! 1          ! SINGLE REPORT FORMAT                                       !
!          !          ! - One 'END OF REPORT' line is produced.                   !
!          !          ! - The transaction 'INPUT NUMBER' is                       !
!          !          !   simply incremented by one for each                       !
!          !          !   transaction.                                             !
!          ! 2          ! SIGN-ON / USERID FORMAT 1                                  !
!          !          ! - An 'END OF REPORT' line is produced for                 !
!          !          !   each userid / sign-on record.                            !
!          !          ! - The transaction 'INPUT NUMBER' is reset                 !
!          !          !   to one for each sign-on record.                          !
!          !          !   The sign-on record will appear as                       !
!          !          !   transaction number one.                                  !
!          ! 3          ! SIGN-ON / USERID FORMAT 2                                  !
!          !          ! - An 'END OF REPORT' line is produced for                 !
!          !          !   each userid / sign-on record.                            !
!          !          ! - The transaction 'INPUT NUMBER' is reset                 !
!          !          !   to zero for each sign-on record.                        !
!          !          !   The sign-on record will appear as                       !
!          !          !   transaction number zero.                                 !
+-----+-----+-----+
```

If the parameter line is not entered, '\$A1' is assumed.

Sign-on line format (required)

```

+-----+
!Col Len! Value ! Description !
+-----+
! 1 1 ! ! ACTION CODE / UPDATE MODE !
! ! ! This field defines the update mode !
! ! ! processing to be used for this userid. !
! ! blank ! NORMAL UPDATE MODE. !
! ! ! - Works like DSMS on-line. !
! ! ! - If an Event or Change is created, all !
! ! ! following sub-screen transactions will !
! ! ! be modified accordingly. !
! ! V ! VERSION CONTROL MODE. !
! ! ! - All batch transactions will be proces- !
! ! ! sed with Action Code 'C' (create). !
! ! ! - The external reference fields on Event !
! ! ! and Change Definitions will be !
! ! ! filled in. !
! ! ! - The associated change fields on Event !
! ! ! Definitions will be converted to the !
! ! ! 'new' Change Number - the number !
! ! ! assigned when the Change is created. !
! ! R ! REORGANIZATION MODE. !
! ! ! - The same as 'V' except that the !
! ! ! external reference fields' content !
! ! ! will not be altered. !
! ! ! !
! 2 1 ! * ! SIGN-ON RECORD CODE !
! 3 8 ! ... ! DSMS USER !
! 11 8 ! ... ! DSMS uSER PASSWORD !
! 19 3 ! ppp ! PRODUCT CODE to which updates apply. !
! 22 2 ! ss ! SUBSIDIARY CODE to which batch updates !
! ! ! apply. !
! 24 1 ! blank ! Unused !
! 25 9 ! ! EXTERNAL REFERENCE VALUES !
! ! ! The value of the next three fields is !
! ! ! used to create Event and Change external !
! ! ! references if the update mode is 'V'. !
! 25 4 ! dddA ! - DSMS external Database code !
! 29 3 ! ppp ! - DSMS external Product code !
! 32 2 ! ss ! - DSMS external Subsidiary code !
+-----+

```

```
+-----+
!Col Len! Value ! Description !
+-----+
! 34 1 !      ! BLANK LINE AFTER ERROR INDICATOR !
!      ! blank ! A blank line is printed after each error !
!      !      ! message on the report. !
!      ! N      ! Blank lines are not printed after error !
!      !      ! messages on the report. !
! 35 1 !      ! REPORT PAGE BREAK INDICATOR !
!      ! blank ! A new page begins only when the number of !
!      !      ! lines per page exceeds the maximum number !
!      ! T      ! A page skip for each new type of trans- !
!      !      ! action !
!      ! E      ! A new page for each transaction type of !
!      !      ! each entity !
! 36 1 !      ! TRANSACTION SORT INDICATOR !
!      ! blank ! The transactions are sorted by type !
!      !      ! before they are processed. !
!      ! N      ! The transactions are processed in their !
!      !      ! arrival order. !
+-----+
```

REPORT

The printout generated by this procedure is an update report, with comments about irregularities or inconsistencies encountered during execution.

RESULT

The result of this procedure is:

- . A DSMS database ready for on-line or batch processing,
- . A Journal file of the transactions which have modified the database, if 'journalization' was not inhibited during the last restoration.

NOTE: This procedure increments the session number if it is the first access to the database for the current day.

13.3. DESCRIPTION OF STEPS

DUPT: DESCRIPTION OF STEPS

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE CONSISTENCY CHECK: PDSBAS

.Permanent input files:

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF: DSN=&INDEX..&ROOT.&FILE.D1

-Error message file

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

.Output report

-Validity report

PACDRS

Return code

-This utility sends a return code 4 and causes an ABEND
in case of database invalidity.

UPDATE OF THE DSMS DATABASE: PDSUP0

.Permanent input-output files

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF : DSN=&INDUV..&ROOT.&FILE.D1

-Va Pac element file

PAC7DC\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DC

PAC7D3\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D3

-Cross-reference file

PAC7DX\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DX

.Permanent input file

-Error message file

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

.Input transaction file

-Update transactions obtained via
the DEXP procedure

PACDIM : DSN=&INPUT

.Output file

-Journal file

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

.Output report

-Update review

PACDRP

.Sort files:

SORTWK01, SORTWK02, SORTWK03

.Return codes

- 0: No error

-08: Error on the user line code or parameter line

-12: I/O error on a file.

13.4. EXECUTION JCL

```

//*****
//* DSMS 2.5 *
//* - UPDATE OF THE DSMS DATABASE - *
//*****
//$RADP.DUPT PROC FILE=$FILE, NUMBER OF PHYSICAL BASE
// ROOT=$ROOT, ROOT OF DSMS SYSTEM
// INDEX='$INDEX', VSAM INDEX
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// CYL='(4,1)', SORT WORKS SIZE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT', SORT LIBRARY
// INPUT='NULLFILE', INPUT TRANSACTIONS DSN (DEXP,DEXT)
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// DBDLIO='$DBDLIO', LIBRARY OF VA PAC DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR
//*
//PDSBAS EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,PDSBAS,PDSBAS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDRS DD SYSOUT=&OUT

```

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


VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
FILE INITIALIZATION

(DINI)

PAGE 113

14

14. FILE INITIALIZATION

(DINI)

14.1. INTRODUCTION

DINI: INTRODUCTION

The DINI procedure initializes the files needed for the installation of a new DSMS database.

It provides an initial backup of the DSMS files, which must be loaded by the Database Restoration (DRST) procedure.

EXECUTION CONDITIONS

None.

However, the parameters of the new DSMS database must have been previously defined, and must be different from the parameters in any other existing DSMS database.

The initial allocation and loading of DSMS components must have been executed (see the Installation Process).

ABNORMAL EXECUTION

Refer to Subchapter 'Abnormal Execution' in Chapter THE BATCH PROCEDURES.

Whatever the cause of the abend, the procedure can be restarted as it is after the problem has been solved.

14.2. INPUT - PROCESSING - RESULTS

USER INPUT

The structure of the input is as follows:

```
+-----+-----+-----+-----+
! POS.! LEN.! VALUE ! MEANING !
+-----+-----+-----+-----+
!  2  !  1  ! 'I'   ! Line code !
!  3  !  1  ! 'l'   ! Initial language code !
!      !     !      ! (E by default: English) !
!  4  !  1  !      ! This field is ONLY used with DOS/VSE !
!      !     ! 'I'   ! Default option for all hardware !
!      !     ! 'N'   ! If CURRENT-DATE = DD/MM/YY in DOS/VSE!
+-----+-----+-----+-----+
```

REPORT

This procedure prints a report listing the memorized options and the number of initial records of the DSMS database files.

RESULT

The result is an initial backup including:

- an initial user, whose userid is '*****' and whose password is '*****' (See the paragraph that follows: INITIAL CONNECTION.)
- a record in the Language Table corresponding to the language code indicated in the user input.

* IMPORTANT NOTE *

INITIAL CONNECTION:

The Database Restoration (DRST) procedure must be executed after the DINI procedure. After a successful execution of the DRST procedure, the DSMS database is installed.

Verify that the on-line access to the new DSMS database is operational.

The initial connection to the DSMS database is executed as follows:

- Access the DSMS database.
- On the Sign-on screen, enter '*****' as the user code and '*****' as the password, then press the ENTER key.
- Among the choices listed on the menu, only those marked with a '*' may be accessed. They correspond to the Tables which must be updated for a proper operation of DSMS. The information must be entered in the Tables in the following order:
 - . In the Languages Table (CHOICE: 'TLA'): the codes and labels of the languages used.
 - . In the Products Table (CHOICE: 'TPR'): the product codes and labels.
 - . In the Subsidiaries Table (CHOICE: 'TSU'): the subsidiary codes and labels.
 - . In the User Parameters Tables (CHOICES: 'TUD', 'TUG', 'TUP' and 'TUS'): user codes and authorizations.

(For more details on the management of these tables, see the DSMS Reference Manual).

The '*****' user code cannot be deleted: after the User Parameters Tables are updated, the DSMS Database Manager should change passwords in order to prevent the use of this code by others.

14.3. DESCRIPTION OF STEPS

DINI: DESCRIPTION OF STEPS

This procedure includes the following steps:

INPUT RECOGNITION: PTU001

INITIAL DATABASE BACKUP: PDSINI

.Input transaction file:

-Initialization transaction
PACDMB : DSN=&&DINIMB

.Permanent input file:

-Error messages
PAC7DESSUF : DSN=&INDEX..&ROOT.&ROOT.DE

.Output file:

-Sequential images of files
PACDDB : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

.Output file:

-Backup report
PACDRU

14.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - INITIALIZATION OF THE DSMS DATABASE -
//*****
//$RADP.DINI PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, DSMS SYSTEM ROOT
// INDEX='$INDEX', VSAM INDEX
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
// INDEXQ='$INDEXQ', DATA GROUP FILE INDEX
// OUT='$OUT', OUTPUT CLASS
// VOLS='SER=$VOLO', GENERATION-FILE VOLUME
// UNITS='$UNITO', GENERATION-FILE UNIT
// SPABB='TRK,(10,2)', BACKUP SPACE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// UWK=$UWK, WORK UNIT
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* INPUT
//* COL 02 : I
//* COL 03 : INITIAL LANGUAGE CODE (F=FRENCH, E=ENGLISH)
//* COL 04 : MACHINE DATE FORMAT (I --> FOR MM/DD/YY)
//* : (N --> FOR DD/MM/YY)
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&DINIMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//PDSINI EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSINI,PDSINI$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//SYSPRINT DD SYSOUT=&OUT
//PACDMB DD DSN=&&DINIMB,DISP=(OLD,PASS)
//PACDDB DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=(&SPABB,RLSE),
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB
//PACDRU DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//*

```

VisualAge Pacbase - Operations Manual	PAGE	119
DSMS - INSTALLATION & OPERATIONS		
JOURNAL EXTRACTION FOR UPDATE	(DXBJ)	15

15. JOURNAL EXTRACTION FOR UPDATE (DXBJ)

15.1. INTRODUCTION

DXBJ: INTRODUCTION

The DXBJ procedure extracts, from the DSMS journal file, all the transactions corresponding to a date/time interval or to a given user, and transforms them into update transactions.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Refer to Chapter THE BATCH PROCEDURES, Subchapter 'Abnormal Execution'.

Whatever the cause of theabend, the procedure can be restarted as it is once the problem has been solved.

15.2. INPUT - PROCESSING - RESULTS

USER INPUT

One '*'-line is required:

```
-----  
!Pos.! Len.! Value  ! Meaning  
!-----+-----+-----!  
! 2 ! 1 ! '*'      ! line code  
! 3 ! 8 ! uuuuuuuu ! DSMS User code  
! 11 ! 8 ! pppppppp ! User password  
!-----+-----+-----!  
!                               ! Optional  
!-----+-----+-----!  
! 19 ! 3 ! ppp      ! Product code  
! 22 ! 2 ! su       ! Subsidiary code  
! 24 ! 1 ! 'F' or 'E' ! Language code  
!   !   !         ! USERS/PASSWORDS IN OUTPUT TRANSAC.  
!-----+-----+-----!
```

One line per extraction request:

```
-----  
!Pos.! Len.! Value  ! Meaning  
!-----+-----+-----!  
! 2 ! 1 ! 'K'      ! Line code  
! 3 ! 1 ! ' '      ! List of selected transactions  
!   !   ! 'N'      ! No list  
! 4 ! 8 ! CCYYMMDD ! Starting date for selection  
! 12 ! 8 ! CCYYMMDD ! Ending date for selection  
! 20 ! 6 ! HHMMSS   ! Starting time for selection  
! 26 ! 6 ! HHMMSS   ! Ending time for selection  
! 32 ! 8 ! uuuuuuuu ! Selected user code  
! 40 ! 1 ! ' '      ! User codes present in journal file  
!   !   !         ! without password.  
!   !   ! 'T'      ! User codes present in journal file  
!   !   !         ! with passwords if sufficient  
!   !   !         ! authorization.  
!   !   ! '1'      ! User code and password, detailed in  
!   !   !         ! following columns.  
! 41 ! 8 ! uuuuuuuu ! User code for output transactions  
!   !   !         ! (if column 40 = 1)  
! 48 ! 8 ! mmmmmmmm ! Password for output transactions  
!   !   !         ! (if column 40 = 1)  
!-----+-----+-----!
```

REPORT

Extraction report and, upon request, the list of formatted transactions.

RESULT

A DSMS update transactions file to be used as input to the DUPT procedure. An 'N' is entered in column 36 of the user line for DUPT not to sort these transactions. .

15.3. DESCRIPTION OF STEPS

DXBJ: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

TRANSACTION EXTRACTION AND FORMATTING: PDS700

.Permanent input files:

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

-Error message file

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

-Archived DSMS journal

PACDBJ : DSN=&INDEXQ..&ROOT.&FILE.BJ

.Input transaction file :

-User transactions

PACDMB : DSN=&&DXBJMB

.Output file:

-Update transaction file for DUPT

PACDIM : DSN=&&PACDIM

.Output reports:

-Extraction review

PACDRK

-Transaction printout

PACDSK

.Return codes:

- 0: No error

- 8: Error on the user '*' line or parameter line.
The environment definition is missing.

-12: File access error.
The technical record is missing.

15.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - EXTRACTION FROM DSMS JOURNAL -
//*****
//$RADP.DXBJ PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, ROOT OF DSMS SYSTEM
// INDEX='$INDEX', VSAM INDEX
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// INDEXP='$INDEXP', INDEX NON VSAM FILES
// INDEQ='$INDEQ', INDEX DATA GROUP FILES
// SPAIM=(TRK,(15,5)), SPACE OF EXTRACTED TRANSACTIONS
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* FORMAT OF TRANSACTIONS AT INPUT :
//* .. A DSMS USER AND PASSWORD LINE
//* COL 02 : *
//* COL 03 : DSMS USER CODE
//* COL 11 : PASSWORD
//* COL 19 : PRODUCT CODE (OPTIONAL)
//* COL 22 : SUBSIDIARY CODE (OPTIONAL)
//* COL 24 : LANGUAGE (OPTIONAL)
//* .. COMMAND LINE(S) FOR EXTRACTION
//* COL 02 : K
//* COL 03 : ' ' SELECTED TRANSACTIONS LIST
//* : 'N' NO LIST OF SELECTED TRANSACTIONS
//* COL 04-11 : STARTING DATE (CCYYMMDD)
//* COL 12-19 : ENDING DATE (CCYYMMDD)
//* COL 20-25 : STARTING HOUR (HHMMSS)
//* COL 26-31 : ENDING HOUR (HHMMSS)
//* COL 32-39 : USER CODE
//* COL 40 : ' ' USER CODES PRESENT IN JOURNAL FILE WITHOUT
//* : PASSWORD
//* : 'T' USER CODE PRESENT IN JOURNAL FILE WITH PASSWORD
//* : IF SUFFICIENT AUTHORIZATION
//* : '1' USER CODE AND PASSWORD, DETAILED IN FOLLOWING
//* : COLUMNS
//* COL 41-47 : USER CODE FOR OUTPUT TRANSACTIONS (IF COLUMN 40=1)
//* COL 48-55 : PASSWORD FOR OUTPUT TRANSACTIONS (IF COLUMN 40=1)
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&DXBJMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDS700 EXEC PGM=DFSRR00,REGION=$REGSIZ,

```

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

VisualAge Pacbase - Operations Manual	PAGE	125
DSMS - INSTALLATION & OPERATIONS		
CODE AND KEYWORD UPDATE	(DREN)	16

16. CODE AND KEYWORD UPDATE (DREN)

16.1. INTRODUCTION

DREN: INTRODUCTION

The Code and Keyword Update procedure (DREN) is used to define new codes (table or site) or new keywords to replace those defined and used until then in the tables, thesaurus, and entities.

EXECUTION CONDITION

This procedure works from a sequential backup and/or an archived journal, and must therefore be preceded by a backup and/or an archiving.

ABNORMAL EXECUTION

See Subchapter 'Abnormal Execution', in Chapter THE BATCH PROCEDURES.

Whatever the cause of theabend, the procedure can be restarted as it is once the problem has been solved.

16.2. INPUT - PROCESSING - RESULTS

USER INPUT

One '*' line (required):

Col.	Len.	Value	Meaning
2	1	*	Line code
3	8	uuuuuuuu	DSMS User Code
11	8	pppppppp	Password
Optional			
19	3	ppp	Changes made on the entities which depend on the product code 'ppp'
		****	Changes made on the entities which depend on all the product codes
22	2	ss	Changes made on the entities which depend on the subsidiary code 'ss'
		***	Changes made on the entities which depend on all the subsidiary codes
24	1	'E' or 'F'	Language code
REQUIRED: AT LEAST ONE OF THESE AREAS SET TO '1'			
25	1	' '	No change concerning the backup
		'1'	Changes concerning the backup
26	1	' '	No change concerning the archiving
		'1'	Changes concerning the archiving

Command lines (500 maxi)

```

+-----+-----+-----+-----+
!Col.! Len.! Value ! Meaning !
+-----+-----+-----+-----+
! 2 ! 3 ! 'Txx' ! table choice (idem TP) !
! ! ! 'Kxx' ! keyword choice (with xx = 'T ' for !
! ! ! ! technical keywords, xx = 'E ' for !
! ! ! ! native keywords and xx = 'Cl' for !
! ! ! ! keywords of change 1 language) !
! ! ! 'S ' ! site choice !
! 5 ! 13 ! ! old code !
! 18 ! 13 ! ! new code !
+-----+-----+-----+-----+
  
```

NOTES:

- The codes (old and new) must be preceded by 'C', 'E' or 'S' for the TST table, by 'C' or 'E' for the TGR and TTY tables, and by 'F' or 'R' for the TAT table.
 - It is not possible to invert two codes (for example, change 'AA' to 'BB', and 'BB' to 'AA'). However, it is possible to rename a code (with an unknown one), and to reuse the old code to transform other codes (for example: 'AA' becomes 'BB' while 'CC' and 'DD' become 'AA'; in this case the command AA/BB must be written before CC/AA and DD/AA).
 - The products, subsidiaries or sites new codes must not already exist (in the same subsidiary for a site).
 - The two parts of the site code (9 and 3 characters) cannot be modified separately.
 - For the TVE table, it is possible to ask for the following updates:
 - . Technical release alone
 - . Technical release and release
 - . Technical release, release and hardware
 - . Technical release, release, hardware and version (with or without language code)
 - . Release alone
 - . Hardware alone
 - . Version number (with or without language code)
- Isolated parts should be aligned as if the other parts were there.

Ascending consistency checks are performed. The changes requested on the preceding lines must be taken into account.

- The label associated to the new code can either be that of the old code or that of the new code if it already existed. This choice is made while the file is sorted and is therefore unpredictable.
- For tables depending on a product (TOP, TPH and TVE), the product's code must be clearly specified on the '*' line.

PRINTED OUTPUT

Report on changes concerning the backup and/or the archiving.

Note on counters:

They count the total number of updates but not the number of modified records (there can be several modifications on the same record).

RESULT

If the change was made on the archive (1 in column 26), a new version of the Journal's sequential backup is produced.

If the change was made on the Database backup (1 in column 25), the result is a new version of the Database sequential backup which should be reorganized via the DREO procedure before being restored.

RETURN CODE

```
+-----+
!  0  ! OK                                     !
!  8  ! Error on the '*' line or on a command line !
! 10  ! Absence or invalid value for backup 'top'  !
! 11  ! Absence or invalid value for archive 'top' !
! 12  ! Input/Output error or inconsistent DSMS base !
!    ! Invalid absence of backup/archive 'tops'  !
! 16  ! Sort error                               !
+-----+
```

16.3. DESCRIPTION OF STEPS

DREN: DESCRIPTION OF STEPS

This procedure calls a single program (PDSMS) which is used as a flow monitor for various programs considered as sub-routines of this monitor. It includes the following steps:

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

UPDATES: PDSMS

.Permanent input files:

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

-Error message file

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

-Crossed reference

PAC7DX\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DX

-DSMS backup

PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(0)

-DSMS archiving

PACDBJ : DSN=&INDEXQ..&ROOT.&FILE.BJ(0)

.Input file:

-User queries

PACDMB : DSN=&&PACDMB

.Work files:

-Update requests

PACDW0

-Partial backup (sorted)

PACDW1

-Partial backup (not sorted)

PACDW2

.Output files:

-Modified backup

PACDB3

-Modified archive

PACDJB

.Output reports:

-Branching report

PACDIA

-List of commands on the backup

PACDIK

-Update report (backup)

PACDJK

-Merging report (backup)

PACDIS

-List of commands on archiving

PACDKK

-Update report (archive)

PACDLK

.Sort files:

SORTWK01, SORTWK02, SORTWK03

16.4. EXECUTION JCL

```

//*****
//* DSMS 2.5 *
//* - CHANGE OF TABLE AND SITE CODES, AND KEYWORDS - *
//*****
//$RADP.DREN PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, ROOT OF DSMS SYSTEM
// LNG='E', LANGUAGE OF MONITOR
// INDEX='$INDEX', VSAM INDEX
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
// INDEXQ='$INDEXQ', INDEX OF GENERATION FILES
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// SPAWK='(TRK,(60,15))', SPACE OF EXTRACTION COMMANDS
// SPABB='(TRK,(20,5),RLSE)', SPACE OF BACKUP (IF DISK)
// SPABJ='(TRK,(20,5),RLSE)', SPACE OF BACKUP (IF DISK)
// CYL=3, SORT WORKS SIZE
// VOL$='SER=$VOLO', GENERATION FILE VOLUME
// UNITS='$UNITS', GENERATION FILE UNIT
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT', SORT LIBRARY
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-----*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PACDMB,DISP=(,PASS),
// UNIT=UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//PDSMS EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,PDSMS&LNG,PDSMS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)

```

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

VisualAge Pacbase - Operations Manual	PAGE	133
DSMS - INSTALLATION & OPERATIONS		
PRE-PROCESSING OF GENERATED DAF PROGRAMS (DPDF)		17

17. PRE-PROCESSING OF GENERATED DAF PROGRAMS (DPDF)

17.1. INTRODUCTION

DPDF: DAF PRE-PROCESSOR FOR GENERATED PROGRAMS

The DPDF procedure processes user generated programs that contain SQL requests for Database access through DAF operators.

EXECUTION CONDITION

None.

IMPLEMENTATION

The DPDF procedure may be executed in several ways:

- Either after a program generation via GPRT, whose generated output is used as input for DPDF, before being passed on for compilation or storing in a source-program library.
- Or by a call in the optional generated program before/after control cards. In this case, the correct JCL must have been entered in the selected options, which are updated by the user-parameter update transaction or the PARM batch procedure.

17.2. INPUT - PROCESSING - RESULTS

USER INPUT

It is the COBOL source of the programs containing DAF operators which must be solved by the pre-processor before being compiled.

Each program contains, after the IDENTIFICATION DIVISION line, a command line for the pre-processor:

```

-----
!Pos.! Len.! Value      ! Meaning
!-----+-----+-----+-----!
! 1 ! 6 ! nnnnnn ! COBOL line number
! 7 ! 1 ! '*' ! Comments
! 8 ! 5 ! 'TP ' ! On-line program, or
! ! ! 'BATCH' ! Batch program
! 13 ! 6 ! 'LIB:' ! Fixed label
! 19 ! 3 ! bbb ! Library code
! 22 ! 1 ! blank ! Not used
! 23 ! 5 ! nnnns ! Session number - Session status
! 28 ! 1 ! blank ! Not used
! 29 ! 2 ! -- ! Generation variant(s)
! 31 ! 5 ! 'AR:' ! Fixed label
! 36 ! 1 ! l ! Database language code
! 37 ! 5 ! 'SC:' ! Batch language program skeleton
! ! ! 'SG:' ! OLSD program skeleton
! ! ! 'SR:' ! COBOL Generator program skeleton
! 42 ! 1 ! l ! Skeleton language
! 43 ! 1 ! blank ! Not used
! 44 ! 6 ! 'SINGLE' ! Single quotes, or
! ! ! 'DOUBLE' ! Double quotes
-----

```

Examples:

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

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

This line is automatically generated by the GPRT procedure.

PRINTED OUTPUT

The procedure prints the list of errors, if any.

RESULT

The result of the execution is a COBOL source file in which all DAF operators have been solved, and all the calls to Database batch or on-line access routines have been generated.

17.3. DESCRIPTION OF STEPS

DPDF: DESCRIPTION OF STEPS

The DPDF procedure calls a single program which acts as a flow monitor for various programs, considered as sub-programs of this monitor. It includes the following step:

INPUT RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

GENERATED PROGRAM'S PRE-PROCESSOR: DAFD10

.Permanent input files:

-Data file

PAC7DA\$\$SUF : DSN=&INDEX..&ROOT.&FILE.DA

PAC7D1\$\$SUF : DSN=&INDEX..&ROOT.&FILE.D1

-Error message file

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

.Input file:

-Generated programs

DAF80 : DSN=&&DPDFMB

.Output files:

-Generated programs to be compiled

COB80 : DSN=&&DAFGEN

.Output reports:

-Execution report

DAFREP

NOTE: If the generated stream contains the compilation's control cards, in the case where DPDF is chained after GPRT, the DSN (&&DAFGEN) may be replaced by a transmission of SYSOUT=(&OUT,INTRDR) to the MVS machine's Internal Reader.

17.4. EXECUTION JCL

```

//*****
//* DSMS 2.5 *
//* - ACCESS FACILITY PRE-PROCESSING - *
//*****
//$RADP.DPDF PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF DSMS SYSTEM
// INDEX='$INDEX', VSAM INDEX
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
// SPAMB='(TRK,(150,15))', SPACE OF GENERATED PROGRAMS
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&DPPDFMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
//*
//DAFD10 EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,DAFD10,PDSDAF$$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//DAF80 DD DSN=&DPPDFMB,DISP=(OLD,DELETE)
//COB80 DD DSN=&DAFGEN,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAMB,

```

PRE-PROCESSING OF GENERATED DAF PROGRAMS (DPDF)
EXECUTION JCL

PAGE

138

17
4

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

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
BATCH UPDATE FROM DAF TABLES

(DUPD)

PAGE 139

18

18. BATCH UPDATE FROM DAF TABLES (DUPD)

18.1. INTRODUCTION

DUPD: INTRODUCTION

The DUPD procedure performs a batch update on the DSMS Database from a sequential file mirroring the DAF tables.

Its operating principle is quite similar to that of the DUPT procedure, except for the format of the input transactions.

EXECUTION CONDITION

Refer to the chapter dedicated to DUPT.

ABNORMAL EXECUTION

Refer to the chapter dedicated to DUPT.

18.2. INPUT - PROCESSING - RESULT

DUPD: INPUT-PROCESSING-RESULTS

USER INPUT

The sequential file of input transactions is produced by a DAF extractor program. Its records mirror the DAF tables (described in the DAF TABLES Manual).

```
-----  
! Pos. ! Length ! Meaning !  
!-----!  
! 1 ! 1 ! Transaction code (C, M, X, D or A, B) !  
! 2 ! 10 ! DAF table code !  
! 12 ! 299 ! DAF table contents (described in the !  
! ! ! DAF tables Manual). !  
-----
```

UPDATE RULES

Update transactions are not sorted.

Each set of transactions impacting a library or session must be preceded by an ASSIGN table code line.

```
-----  
! Pos. ! Length ! Value ! Meaning !  
!-----!  
! 2 ! 10 ! 'ASSIGN' ! Table code !  
! 12 ! 8 ! uuuuuuuu ! User code !  
! 20 ! 8 ! pppppppp ! Password !  
! 28 ! 3 ! ppp ! Product code !  
! 31 ! 2 ! ss ! Subsidiary code !  
-----
```

PRINTED OUTPUT

Refer to the description of the DUPT output.

RESULT

Refer to the description of the DUPT result.

18.3. DESCRIPTION OF STEPS

DUPD: DESCRIPTION OF STEPS

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE CONSISTENCY CHECK: PDSBAS

.Permanent input files:
-Data file
PAC7DA\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$SUF: DSN=&INDEX..&ROOT.&FILE.D1
-Error message file
PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

.Output report
-Validity report
PACDRS

Return code
-This utility sends a return code 4 and causes an ABEND
in case of database invalidity.

TRANSACTION FORMATTING: PDS900

.Input transaction file:
-Update transactions
PACDGY: DSN=&DAFINPUT Length=382

.Output files:
-Formatted transactions
PACDIM: DSN=&&PACDIM Length=250

DSMS SATABASE UPDATE : PDSUP0

. Permanent input-output files :

- Input files
PAC7DA\$SUF : DSN=&INDEX..&ROOT.&FILE.DA
PAC7D1\$SUF : DSN=&INDEX..&ROOT.&FILE.D1
- VA Pac elements file
PAC7DC\$SUF : DSN=&INDEX..&ROOT.&FILE.DC
PAC7D3\$SUF : DSN=&INDEX..&ROOT.&FILE.D3
- Cross-references file
PAC7DX\$SUF : DSN=&INDEX..&ROOT.&FILE.DX

. Permanent input file

- Error message file
PAC7DE\$SUF : DSN=&INDEX..&ROOT.&ROOT.DE

. Input transaction file

- update transactions
PACDIM : DSN=&&PACDIM

. Output files

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

. Output reports

- update report
PACDRP

. Sort files

BATCH UPDATE FROM DAF TABLES
DESCRIPTION OF STEPS

(DUPD)

PAGE

143

18
3

SORTWK01, SORTWK02, SORTWK03

. Return code

- . 0 : No error detected on files
- .08 : Error on the user '*' line or parameter line
- .12 : input-output error on a file

18.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - UPDATE OF THE DSMS DATABASE - DAF
//*****
//$RADP.DUPT PROC FILE=$FILE, NUMBER OF PHYSICAL BASE
// ROOT=$ROOT, ROOT OF DSMS SYSTEM
// INDEX='$INDEX', VSAM INDEX
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
//*: SYSTCAT='$CATV', VSAM SYSTEM CATALOG
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// CYL='(4,1)', SORT WORKS SIZE
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT', SORT LIBRARY
// DAFINPUT='NULLFILE', DSN UPDATE MOUVEMENT
// SPAIM='(TRK,(100,10),RLSE)', TRANSACTION SPACE
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// DBDLIO='$DBDLIO', LIBRARY OF VA PAC DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//DDDC DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//DDD3 DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//DDDE DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//DDDX DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDC),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD3),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDE),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDX),DISP=SHR
//*
//PDSBAS EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSBAS,PDSBAS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDR DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR

```



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

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
INSTALLATION

PAGE 146

19

19. INSTALLATION

19.1. ENVIRONMENT PREPARATION

ENVIRONMENT PREPARATION

The site must be prepared for the installation of the DSMS system, taking into account its technical characteristics, as described in the initial chapters of this manual:

- . Select the VA Pac suffixes, prefixes and roots to be assigned to DSMS files and codes,
- . Select the VSAM catalogs and allocate adequate disk space,
- . And, above all, prepare the IMS/DC generation by taking into account the parameters identified below.

PARAMETERS FOR IMS/DC GENERATION

In the following names, the symbols yy, zz and rr represent two suffixes and a radical chosen by the user. The installation is easier if the following values are selected (in which case, the compilation of DBDs, PSBs and FORMATS is not necessary):

```
zz = 22 : for the suffix of the batch PSBs,  
yy = 22 : for the suffix of the DBDs,  
rr = P2 : for the prefix of the on-line PSBs,  
and transactions.  
(NOTE: prefix = system root)
```

1. Declaration of the DBDs used:

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

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

(1) These DBDs are supplied with the VA Pac system.

2. Declaration of the DSMS transaction codes:
(Conversational)

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

 where: tttttttt = the transaction code selected
 for site connection to DSMS (1 to 8 characters).

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

NOTE

It is strongly recommended that DSMS transactions be used in
RESPONSE mode.

Parameter sources for IMS generation are supplied on the
tape in the System Parameters PDS.

19.2. INTRODUCTION

INTRODUCTION

The installation procedure breaks down into three main phases:

- . Preparation for installation,
- . Installation,
- . On-line and batch tests.

The procedure uses an installation cartridge (or tape). The complete installation process is described in this chapter.

Before proceeding with the actual installation, it is important that the user be familiar the technical characteristics of the DSMS function described in this manual. This information is needed to prepare the required environment for the installation procedure (disk space, various catalogs, etc.).

Once the environment is prepared, the installation can take place.

PREPARATION

Retrieval of the initial JCL from the tape and execution of this JCL:

- . Backup of the installation tape,
- . Copy complete JCL into a processing module,
- . Retrieval of the complete DSMS installation and operation JCL.

ACTUAL INSTALLATION

See Subchapter 'INSTALLATION PROCESS'.

TESTING

- . On-line tests,
- . Batch tests.

19.3. INSTALLATION CARDRIDGE (OR TAPE)

CONTENTS OF INSTALLATION TAPE

The installation tape (6,250 BPI, standard labels) contains the following files:

```

-----
!RANK! LABEL          ! LRECL ! BLKS ! CONTENTS          !
!-----!-----!-----!-----!-----!
! 01 ! INST.JCL           !   80  !11,440! Initial preparation !
!   !                   !      !      ! JCL                !
!   !                   !      !      !                    !
! 02 ! INST.MOD           !      ! 6,144! Load module MM1JCL, !
!   !                   !      !      ! JCL preparation uti- !
!   !                   !      !      ! lity.              !
!   !                   !      !      !                    !
! 03 ! PACD.JCL           !   80  !11,440! Skeleton JCL for    !
!   !                   !      !      ! installation and    !
!   !                   !      !      ! operation           !
!   !                   !      !      !                    !
! 04 ! PACD.MBR8          !      !6,144 ! Batch load modules  !
!   !                   !      !      !                    !
! 05 ! PACD.MTR8          !      !6,144 ! On-line load modules !
!   !                   !      !      !                    !
! 06 ! PACD.DBDLIB        !      !6,144 ! Object DBD's file   !
!   !                   !      !      !                    !
! 07 ! PACD.PSBLIB        !      !6,144 ! Object PSB's file   !
!   !                   !      !      !                    !
! 08 ! PAC.BB             !  354 !6,376 ! Test database backup !
!   !                   ! (VB) !      !                    !
! 09 ! PAC.DE             !   90 !6,300 ! Error message system !
!   !                   !      !      ! file               !
!   !                   !      !      !                    !
! 10 ! PACD.SOURCE        !   80 !6,080 ! Source files for    !
!   !                   !      !      ! user check and DAF  !
!   !                   !      !      ! dictionary          !
! 11 ! PACD.README        !  133 !5,320 ! README file + infor- !
!   !                   ! (FBA) !      ! mation on release   !
!   !                   !      !      !                    !
-----

```

19.4. INSTALLATION PREPARATION

INSTALLATION PREPARATION

It is recommended to copy all the DSMS preparation, installation, and operation JCLs in one special PDS file.

The first installation step is therefore the allocation of this PDS file, whose characteristics are the following:

- Lrecl=80
- Size: about 30 tracks of a 3,390 disk, and 20 directory blocks.

Note: This allocation must be performed by the person in charge of the installation.

The second step is the copy of the DSMS installation tape's or cartridge's initial JCL ('INST.JCL') by one of the site's utilities (i.e. IEBGENER), in the PDS previously allocated.

INITIAL JCL

The purpose of the Initial JCL ('INST.JCL') is to generate the installation and operation JCL of the DSMS function. It is provided as a parameterized skeleton which can be adapted to the specific needs of each site.

This skeleton is processed by the 'MM1JCL' utility, which generates an adapted JCL, using the appropriate designated parameters.

This utility is unloaded and used in the Initial JCL, which contains three JOBS:

- . COPY of the installation cartridge (or tape) provided by IBM onto a user cartridge (or tape): this constitutes the DSMS backup and it must be used for the actual installation.
- . UNLOAD of the MM1JCL utility stored in the INST.MOD file, using IEBCOPY into a library which already exists on the site or specially dedicated to this operation. This operation is only executed for the first installation of the system.
- . EXECUTION of the MM1JCL to install the DSMS installation and operation JCL. The output from this execution should be saved; it can be used again for re-installation.

The JCL lines should be completed as follows:

```
//STEPLIB DD DSN= <-- library containing MM1JCL
//SYSUT1 <-- specify cartridge number

//SYSUT2 DD DSN= <-- target file for the complete
                    installation-operations JCL.
                    This file can be either a member
                    of the PDS allocated for all the
                    JCLs, or a sequential file
                    chosen by the user.
```

Enter the parameters (see following subchapters).

THIS EXECUTION MUST BE KEPT: IT CAN BE RE-USED FOR RE-INSTALLATIONS.

INSTALLATION
 INSTALLATION PREPARATION
 INITIAL JCL

19
 4
 1

19.4.1. INITIAL JCL

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

INSTALLATION

19

INSTALLATION PREPARATION

4

INITIAL JCL

1

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


INSTALLATION
INSTALLATION PREPARATION
INITIAL JCL

PAGE

156

19
4
1

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

DEDIM000252A

INSTALLATION

19

INSTALLATION PREPARATION

4

INITIAL JCL

1

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

INSTALLATION	PAGE	158
INSTALLATION PREPARATION		19
COMPLETE JCL INSTALLATION		4
		2

19.4.2. COMPLETE JCL INSTALLATION

INSTALLATION PREPARATION

INSTALLATION OF THE COMPLETE JCL

The MM1JCL module reads the JCL skeleton file (label 03) and produces a complete JCL. It allows you to:

- . Select the installation language,
- . Select portions of the skeleton JCL, which are called 'JCL modules',
- . Parameterize the skeleton in order to obtain a JCL requiring a minimum of modifications to make it operational,
- . Select the installation variants to generate the JCL needed for specific processing, depending on site and installation conditions,
- . Add lines before and after the JCL modules to separate them.

This step can be executed as many times as necessary to generate a complete JCL.

USER INPUT

Refer to the following paragraphs:

- .Coding of MM1JCL commands
- .Installation variants
- .JCL modules
- .JCL parameters
- .JCL separators

OUTPUT RESULT: COMPLETE JCL

The resulting SYSUT2 file contains all the installation and operation JCLs. This file may be modified (if necessary) via a text editor before beginning the installation.

	PAGE	159
INSTALLATION		19
INSTALLATION PREPARATION		4
COMPLETE JCL INSTALLATION		2

Two operations must be performed on the complete JCL:

1. Global modifications (if necessary):

Adaptations can be performed on all the JCLs.

VSAM catalogues are entered as comments in the installation JCL:

- in the DELETE/DEFINE, as: /*: CATALOG (\$VCAT) */
or: /*: CATALOG (\$SCAT) */
- in the JCL STEPCAT's as: /*: STEPCAT DD
and/or: /*: DD
- in the procedure parameters as: /*: VSAMCAT='\$VCAT'
or: /*: SYSTCAT='\$SCAT'

When these parameters are not required, the resulting JCL can remain as it is.

When these parameters are required, affected lines should be changed into command lines. This is accomplished by:

- Transforming all '/*:' into '//',
- Substituting blanks for '/*:' and '*/'.

Blocking factors for large files can also be changed. Refer to paragraph 'Note on the files' Csize/Blksize' (VA Pac only).

CAUTION: SMS

- . If the SMS product is installed, you should delete IDCAMS definition DD //GDGMOD lines in the installation JCLs with GenerationDataGroup allocation.
- . If the UNIT and VOL parameters cannot be used on the site, you can delete them in the whole JCL through an exclusion (EXCLUDE command of TSO/EDIT).

In most cases, it is recommended to perform general modifications on JCLs before the JCL splitting operation.

	PAGE	160
INSTALLATION		19
INSTALLATION PREPARATION		4
COMPLETE JCL INSTALLATION		2

2. JCL splitting

In front of each module of a standard complete JCL, there is a ./ ADD NAME=<JCL-module> line, where <JCL-module> is the code of the ===MOD line that is found (see the following table of JCL modules).

This allows for the complete JCL to be split in as many members as there are JCL modules in a PDS. The completed JCL file is to be used as SYSIN for the PDS update utility: IEUBUPDTE.

NOTE:

Because of this default option, all './' characters found in JCL modules containing IEUBUPDTE were replaced with './'.

Once the JCL is split, the replacement must be done the other way round before executing jobs which contain IEUBUPDTE.

REPORT

MM1JCL produces a list for each JCL module created, including parameters taken into account and according to required variants.

Note: since the JCL skeleton parameters are in the \$xxxx format, during execution, if MM1JCL encounters a \$ character that does not correspond to a defined parameter, it sends error messages such as: 'UNKNOWN SYMBOLIC PARAMETER' or 'INVALID POSITION OR LENGTH' or 'SYNTAX ERROR IN SYMBOLIC PARAMETER'.

These messages do not stop the execution and should be ignored: they apply to the '\$' character in the flow processed by MM1JCL and which is NOT a parameter.

INSTALLATION	
INSTALLATION PREPARATION	
COMPLETE JCL INSTALLATION	

19
4
2

CODING OF MM1JCL COMMANDS

```

===SELL lll           .Selection of installation
                       language:
                       lll = ENG (English)
                       FRA (French)

===SELV vvvv         .Selection of variant
                       vvvv = variant code

===SELM mmmm1 mmmm2 ... .Selection of JCL modules
                       mmmm1 = name of JCL module
                       mmmm2 = name of JCL module
                       etc.
                       The absence of a ===SELM line
                       involves the selection of all
                       JCL modules.

===PRM  PPPP=pppp    .Parameter
                       PPPP = name of parameter
                       pppp = value of parameter

```

CAUTION: on ===PRM or ===SELV lines, comments may be entered. They should be preceded by a period, and not exceed column 72.

```

===BEGMOD           Insertion of lines at beginning of module.
....1             )
.....            ) Lines to be inserted before each module
....n             )

===ENDMOD           Insertion of lines at end of module.
....1             )
.....            ) Lines to be inserted after each module
....n             )

```

INSTALLATION	
INSTALLATION PREPARATION	
INSTALLATION DEFAULT OPTIONS	

19
4
3

19.4.3. INSTALLATION DEFAULT OPTIONS

DEFAULT INSTALLATION OPTIONS

.VARIANTS (===SELV): all available variants are selected.

IMPORTANT: DELETE THE LINES CORRESPONDING TO THE VARIANTS NOT INSTALLED ON THE SITE OR WHEN THERE IS INCOMPATIBILITY.

.PARAMETERS (===PRM):

Indicated values are examples; they should be replaced according to the site's specific needs.

.MODULES (===SELM):

No selection; all modules (corresponding to the variants) are copied.

.JCL MODULE FIRST LINE (===BEGMOD):

A line: ./ ADD NAME=\$MODULE

This adds a line before each JCL module, in the form:

```
./ ADD NAME=<name-of-JCL-module>
```

INSTALLATION
 INSTALLATION PREPARATION
 INSTALLATION VARIANTS

19
 4
 4

19.4.4. INSTALLATION VARIANTS

TABLE OF VARIANTS

```

===SELV vvv          .comment

+-----+
! vvv ! Meaning          !
+-----+-----+
!     !                  !
! DAF ! DSMS Access Facility !
!     !                  !
! ROLD ! Selection of retrieval of the !
!     ! old DSMS           !
!     !                  !
! SEC  ! Security system interface      !
!     ! Installation of sub-routines    !
!     ! allowing access to the Security !
!     ! System (RACF or TOPSECRET)     !
+-----+-----+

```

INSTALLATION	PAGE	164
INSTALLATION PREPARATION		19
JCL MODULES		4
		5

19.4.5. JCL MODULES

TABLE OF JCL MODULES

===SELM mmmmm1 mmmmm2 ... mmmmmN

! mmmmm	! CONTENTS	! NATURE!
! PACSLOAD	! Load batch (MBR8) & on-line (MTR8)	! JCL !
!	! program libraries	! !
! PACSSY	! Load SY Parameters PDS	! JCL !
! PACSPROC	! Load execution procedures	! JCL !
! PACSDDB	! Load DBDLIB	! JCL !
! PACSPSB	! Load PSBLIB	! JCL !
! PACSPRE	! Initial allocations and loadings	! JCL !
! PACSDE	! Load DSMS error message database	! JCL !
! PACSACB	! Compilation of the ACB's	! JCL !
!	! -----	! !
! PACSRC	! Allocation/loading of sources' PDS for	! JCL !
!	! user check	! !
! PACSDS	! Test deck restoration	! JCL !
!	!	! !

TABLE of JCL MODULES (cont'd)

! mmmm	! Contents	! Nature!
!	!	!
!	! EXAMPLES OF JOBS FOR TESTS	!
!	!	!
! JCLDARC	! Journal archiving	! JCL !
!	!	!
! JCLDEXT	! Extraction (DEXT) and update where applicable (DUPT)	! - !
!	!	!
! JCLDINS	! List of program dates	! - !
!	!	!
! JCLDRST	! Restoration of database	! - !
!	!	!
! JCLDSAV	! Backup of database	! - !
!	!	!
! JCLDXBJ	! Extraction of journal in the form of update transactions	! - !
!	!	!
! JCLDAF	! Execution of a batch program allowing access to DSMS via DAF	! - !
!	!	!

19.4.6. JCL PARAMETERS

TABLE OF PARAMETERS

```

===PRM PPPP=pppp                                .Comments

+-----+-----+-----+-----+
! CODE ! MEANING ! DEFAULT !
! PPPP ! ! ! PPPP !
+-----+-----+-----+-----+
! ! ! !
! ! JOB CARDS ! !
! ! ----- ! !
! ! ! !
!PRFJ ! 3-character prefix for job name ! PAC !
!CCPT ! Job accounting code ! <> !
!CLASSJ ! Job execution class ! 1 !
!MSGCL ! JCL output class ! A !
! ! ! !
! ! DSN CODIFICATION ! !
! ! ----- ! !
! ! ! !
! ! All permanent DSMS files (except ! !
! ! for the load-module libraries) ! !
! ! have names in the following format: ! !
! ! ! !
! ! INDEX.xxxxss : VSAM System ! !
! ! INDEXP.xxxxss : Non-VSAM System ! !
! ! INDEX.xxnns : VSAM User ! !
! ! INDEXQ.xxnns : Non-VSAM User (GDG) ! !
! ! ! !
! ! IND-- Index of file names: ! !
!INDEX ! VSAM System and User (2)!PAC25IMS.IMS!
!INDEXO! Non-VSAM VA Pac System (SAM, PDS) (1)! PAC25.SEQ !
!INDEXP! Non-VSAM System (SAM, PDS) ! PAC25.SEQ !
!INDEXQ! Non-VSAM User (GDG) ! PAC25.SEQ !
! ! ! !
! ! xx=ROOT, nn=FILE, ss=file code ! !
!ROOT ! Root of the DSMS System ! !
! ! (2 characters other than 'ZZ') (2)! P2 !
!ROOTO ! Root of the VA Pac System ! !
! ! (2 characters other than 'ZZ') (1)! P2 !
!FILE ! Number of DSMS user database ! !
! ! (2 digits other than 99) (2)! 00 !
!FILEO ! Number of VA Pac user database ! !
! ! (2 digits other than 99) (1)! 00 !
+-----+-----+-----+-----+

```

TABLE OF PARAMETERS (CONTINUED)

!CODE	! MEANING	! DEFAULT	!
!PPPP	!	! pppp	!
!	!	!	!
!	! DD CARDS	!	!
!	! -----	!	!
!	!	!	!
!OUT	! Sysout print class	! A	!
!UTAPE	! Installation tape UNIT, duplicate	! 3480	!
!	! of initial tape supplied by IBM	!	!
!TAPEI	! Name of installation tape, duplicate	! <>	!
!	! of initial tape supplied by IBM	!	!
!UWK	! UNIT of work files used	! SYSDA	!
!UNITP	! UNIT of system non-VSAM files	! 3390	!
!UNITO	! UNIT of user non-VSAM files	! 3390	!
!UNITV	! UNIT of system VSAM files	! 3390	!
!UNITU	! UNIT of user VSAM files	! 3390	!
!VOLP	! Volume name of non-VSAM system files	! <>	!
!VOLV	! Volume name of VSAM system files	! <>	!
!VOLO	! Volume name of non-VSAM user files	! <>	!
!VOLU	! Volume name of VSAM user files	! <>	!
!	!	!	!

MISCELLANEOUS PARAMETERS

! CODE !	! MEANING	! DEFAULT	!
! PPPP !		! pppp	!
!	! Other parameters	!	!
!RADP	! Prefix names of procedures to be installed (maximum of 4 characters).	! D250	!
!	! This parameter should be changed only when the standard DSMS procedure prefix is not suitable.	!	!
!HEXA	! System root in hexadecimal (see ROOT parameter).	! D7F2	!
!REGSIZ	! Batch procedure region size	! 1536K	!
!DBRC	! Batch procedure DBRC utilization	! N	!
!IRLM	! Batch procedure IRLM utilization	! N	!
!CATU	! DSNAME of VSAM catalog in which the installed DSMS test database is to be catalogued	! <>	!
!CATV	! DSNAME of VSAM catalog in which the DSMS files are to be catalogued	! <>	!
!MODB	! DSNAME of DSMS batch load-module library	! PAC25. ! PGMLIBBH	!
!MODT	! DSNAME of DSMS on-line load-module library	! PAC25. ! PGMLIBTP	!
!BIBP	! DSNAME of the procedure library in which the DSMS procedures are to be catalogued	! SYS1.PROCLIB	!
!BIBT	! DSNAME of the sort library	! SYS1.SORTLIB	!
!PSBLIB	! DSNAME PSB library	! IMSVS.PSBLIB	!
!ACBLIB	! DSNAME ACB library	! IMSVS.ACBLIB	!
!DBDLIB	! DSNAME DBD library	! IMSVS.DBDLIB	!
!DBDLIO	! PACBASE DSNAME DBD library	(1) ! IMSVS.DBDLIB	!
!RESLIB	! DSNAME IMS RESLIB	! IMSVS.RESLIB	!
!PRCLIB	! DSNAME IMS PROCLIB	! IMSVS.PROCLIB	!
!SUF	! Suffix of the DSMS DBD's	(2) ! 22	!
!SUG	! Suffix of batch PSB's	! 22	!
!CLS	! Transaction code class	! 2	!
!TRANS	! Transaction code for connection to DSMS	! DSMS	!

	PAGE	169
INSTALLATION		19
INSTALLATION PREPARATION		4
JCL PARAMETERS		6

- (1) The parameters 'ROOTO', 'FILEO', 'INDEXO', AND 'DBDLIO' MUST have the same values as the VA Pac parameters 'ROOT', 'FILE', 'INDEXP', and 'DBDLIB' already set on the site.
- (2) The parameters 'ROOT', 'FILE', 'SUF', AND 'INDEX' MUST have the same values as the VA Pac parameters 'ROOTD', 'FILED', 'SUFD', and 'INDEXD' already set on the site.

NOTE:

- . '<>' indicates a parameter that must be coded.
- . If a parameter contains special characters, it must be entered between single quotes (').

19.4.7. JCL MODULES SEPARATORS

JCL MODULES SEPARATORS

```
===BEGMOD  
....1   )  
.....   ) Lines to be inserted before each JCL module  
....n   )
```

```
===ENDMOD  
....1   )  
.....   ) Lines to be inserted after each JCL module  
....n   )
```

Lines may be inserted as input in the MM1JCL if the default option is not appropriate (see Subchapter 'INSTALLATION DEFAULT OPTIONS' above).

The purpose of these lines is to execute the separation of the JCL file created by the MM1JCL utility into as many members as there are JCL modules.

This utility adds1 ton lines in front of each JCL module and1 ton lines to the end of each JCL module.

19.5. INSTALLATION PROCESS

THE INSTALLATION PROCESS

The JCL obtained after PHASE 2 is made up of 11 jobs (each job being a JCL module) which constitute the DSMS System installation.

1. Load batch and on-line programs,
2. Load system parameters PDS,
3. Load batch operation procedures,
4. Load DBDLIB,
5. Load PSBLIB,
6. Initial preparation of system files,
7. Loading of the DSMS Error Messages file,
8. Compilation of ACBs,
9. Installation complements,
10. Loading of test database,
11. List of installed programs with compilation dates.

	PAGE	172
INSTALLATION		19
INSTALLATION PROCESS		5
LOADING BATCH AND ON-LINE PROGRAMS		1

19.5.1. LOADING BATCH AND ON-LINE PROGRAMS

1. LOADING BATCH AND ON-LINE PROGRAMS

The load of batch and on-line programs is made up of a JOB '\$PRFJ.LOAD' which contains the following steps:

- ET010 : IEHPROGM : SCRATCH UNCATLG of batch and on-line load-module libraries.
- ET020 : IEFBR14 : allocation of load-module libraries.
- ET030 : IEBCOPY : loading the load-modules.
- ET040 : IMASPZAP : ZAP of the initiator program of the DSMS transaction: \$ROOT.00ZZ.
To be executed only if the value chosen for \$ROOT is not 'P2'.
- ET050 : IEBCOPY : Loading of the RACF-table access routine PASECU8 into an authorized program library.
Definition of a resource class is not necessary since checks are performed on the user code and password only.

(See JCL below.)

INSTALLATION

19

INSTALLATION PROCESS

5

LOADING BATCH AND ON-LINE PROGRAMS

1

```

//$PRFJ.LOAD JOB ($CCPT), 'LOAD-MODULES', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//* --- COPY BATCH AND TP LOAD-MODULES IN PDS --- DSMS *
//*****
//*
//*-----*
//*          I M P O R T A N T
//*          -----*
//* IT IS NOT NECESSARY TO EXECUTE THE 'ET040' STEP *
//* IF THE ON-LINE MODULES PREFIX CONTAINS THE *
//* VALUE 'C1' (CHECK 'ROOT' PARAMETER). *
//* IF THE PREFIX IS MODIFIED, THIS STEP MUST BE *
//* EXECUTED. IN ORDER TO DO SO, THE 'HEXA' *
//* PARAMETER MUST BE CODED WITH THE CORRESPON- *
//* -DING HEXADECIMAL VALUE OF THE NEW PREFIX. *
//* FOR INST: IF THE NEW PREFIX IS: ROOT='AB', *
//* THE 'HEXA' PARAMETER MUST BE *
//* CODED: HEXA='C1C2' *
//* THE 'ET050' STEP ONLY APPLIES TO THE 'SEC' OPTION *
//* (COPY - PACSECU8 -) *
//*-----*
//*
//ET010 EXEC PGM=IEHPRGM
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD UNIT=$UNITP, VOL=SER=$VOLP, DISP=SHR
//SYSIN DD *
        UNCATLG DSNAME=$MODB
        SCRATCH DSNAME=$MODB, VOL=$UNITP=$VOLP
        UNCATLG DSNAME=$MODT
        SCRATCH DSNAME=$MODT, VOL=$UNITP=$VOLP
//*
//ET020 EXEC PGM=IEFBR14
//DDA DD DSN=$MODB, DISP=(,CATLG,DELETE), UNIT=$UNITP,
//     SPACE=(TRK,(55,10,15)), VOL=SER=$VOLP,
//     DCB=(RECFM=U, BLKSIZE=6144)
//DDB DD DSN=$MODT, DISP=(,CATLG,DELETE), UNIT=$UNITP,
//     SPACE=(TRK,(70,10,15)), VOL=SER=$VOLP,
//     DCB=(RECFM=U, BLKSIZE=6144)
//*
//ET030 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3 DD UNIT=$UWK, SPACE=(TRK,20)
//SYSUT4 DD UNIT=$UWK, SPACE=(TRK,20)
//INB DD DSN=PACD.MBR8, DISP=OLD, UNIT=$UTAPE, LABEL=(04,SL),
//     VOL=(,RETAIN,SER=$TAPEI)
//INT DD DSN=PACD.MTR8, DISP=OLD, UNIT=$UTAPE, LABEL=(05,SL),
//     VOL=(,RETAIN,SER=$TAPEI)
//OUTB DD DSN=$MODB, DISP=SHR
//OUTT DD DSN=$MODT, DISP=SHR
//SYSIN DD *
        COPY INDD=( (INB,R) ), OUTDD=OUTB
===SEQ FOR DAF
S M=DAFD10
S M=PDTPDF
===SEQ
S M=PDCHOI
S M=PDSA10
S M=PDSB
S M=PDSEX
S M=PDSMS
S M=PDSBE
S M=PDSEXE
S M=PDSMSE
S M=PDSUQ1
S M=PDSUQ2
S M=PDSUQ5
S M=PDSUQ6
S M=PDSUQ7
S M=PDSUQ8
S M=PDSRCT
S M=PDSRFU
S M=PDSRMS

```

INSTALLATION

19

INSTALLATION PROCESS

5

LOADING BATCH AND ON-LINE PROGRAMS

1

S M=PDSJMS
 S M=PDSCAM
 S M=PDSCFV
 S M=PDSCMQ
 S M=PDSCRQ
 S M=PDSCSI
 S M=PDSCBAS
 S M=PDSE90
 S M=PDSEFAC
 S M=PDSEINI
 S M=PDSELVB
 S M=PDSESRQ0
 S M=PDSESRQ1
 S M=PDSESRQ2
 S M=PDSESRQ3
 S M=PDSESR10
 S M=PDSESR20
 S M=PDSESR30
 S M=PDSESR40
 S M=PDSESUAA
 S M=PDSESUB1
 S M=PDSESUB2
 S M=PDSESUB3
 S M=PDSESUB4
 S M=PDSEUE1
 S M=PDSEUE2
 S M=PDSEUE3
 S M=PDSEUK1
 S M=PDSEUSI
 S M=PDSEUS1
 S M=PDSEUS3
 S M=PDSEUS4
 S M=PDSEUS6
 S M=PDSEUTT
 S M=PDSEUTV
 S M=PDSEUTW
 S M=PDSEUTX
 S M=PDSEUTY
 S M=PDSEUTZ
 S M=PDSEUT1
 S M=PDSEUT2
 S M=PDSEUT3
 S M=PDSEUT4
 S M=PDSEUT5
 S M=PDSEUT6
 S M=PDSEUT7
 S M=PDSEUT8
 S M=PDSEUT9
 S M=PDSEUP0
 S M=PDSEUP1
 S M=PDSEV10
 S M=PDSEV20
 S M=PDSEXCT
 S M=PDSEXDT
 S M=PDSEXST
 S M=PDSEXTH
 S M=PDSEXTR
 S M=PDSE300
 S M=PDSE320
 S M= ((PDSE381 , PDSE380))
 S M=PDSE400
 S M=PDSE450
 S M=PDSE500
 S M=PDSE600
 S M=PDSE610
 S M=PDSE700
 S M=PDSEV090
 S M=PDSE900
 ===SEQ FOR ROLD
 S M=PDSESR8B
 S M=PDSESR8C
 S M=PDSESR8Q
 S M=PDSESR8R

INSTALLATION
INSTALLATION PROCESS
LOADING BATCH AND ON-LINE PROGRAMS

19
5
1

```
S M=PDSR15
S M=PDSR5J
===SEQ
S M=REP2PJ
S M=PTU001
===SEQ FOR SEC
S M=PACSECB
===SEQ
COPY INDD=( (INT,R) ),OUTDD=OUTT
===SEQ FOR SEC
S M=PACSECB
===SEQ
S M=PACX10
S M=PACXAB
S M=MVSJOB
S M=((ZZR980,$ROOT.R980))
S M=((ZZCHOI,$ROOT.CHOI))
S M=((ZZ00AA,$ROOT.00AA))
S M=((ZZ00AB,$ROOT.00AB))
S M=((ZZ00BA,$ROOT.00BA))
S M=((ZZ00B1,$ROOT.00B1))
S M=((ZZ00B2,$ROOT.00B2))
S M=((ZZ00B3,$ROOT.00B3))
S M=((ZZ00B4,$ROOT.00B4))
S M=((ZZ00B5,$ROOT.00B5))
S M=((ZZ00EA,$ROOT.00EA))
S M=((ZZ00E1,$ROOT.00E1))
S M=((ZZ00E2,$ROOT.00E2))
S M=((ZZ00E3,$ROOT.00E3))
S M=((ZZ00E4,$ROOT.00E4))
S M=((ZZ00E5,$ROOT.00E5))
S M=((ZZ00E6,$ROOT.00E6))
S M=((ZZ00FA,$ROOT.00FA))
S M=((ZZ00FB,$ROOT.00FB))
S M=((ZZ00HE,$ROOT.00HE))
S M=((ZZ00JO,$ROOT.00JO))
S M=((ZZ00KA,$ROOT.00KA))
S M=((ZZ00K1,$ROOT.00K1))
S M=((ZZ00K2,$ROOT.00K2))
S M=((ZZ00K3,$ROOT.00K3))
S M=((ZZ00LE,$ROOT.00LE))
S M=((ZZ00LS,$ROOT.00LS))
S M=((ZZ00MA,$ROOT.00MA))
S M=((ZZ00PA,$ROOT.00PA))
S M=((ZZ00P1,$ROOT.00P1))
S M=((ZZ00QA,$ROOT.00QA))
S M=((ZZ00QB,$ROOT.00QB))
S M=((ZZ00QC,$ROOT.00QC))
S M=((ZZ00Q1,$ROOT.00Q1))
S M=((ZZ00Q2,$ROOT.00Q2))
S M=((ZZ00Q3,$ROOT.00Q3))
S M=((ZZ00Q4,$ROOT.00Q4))
S M=((ZZ00Q5,$ROOT.00Q5))
S M=((ZZ00Q6,$ROOT.00Q6))
S M=((ZZ00Q7,$ROOT.00Q7))
S M=((ZZ00Q8,$ROOT.00Q8))
S M=((ZZ00Q9,$ROOT.00Q9))
S M=((ZZ00SA,$ROOT.00SA))
S M=((ZZ00SI,$ROOT.00SI))
S M=((ZZ00S1,$ROOT.00S1))
S M=((ZZ00S3,$ROOT.00S3))
S M=((ZZ00S4,$ROOT.00S4))
S M=((ZZ00S5,$ROOT.00S5))
S M=((ZZ00S6,$ROOT.00S6))
S M=((ZZ00S7,$ROOT.00S7))
S M=((ZZ00S8,$ROOT.00S8))
S M=((ZZ00S9,$ROOT.00S9))
S M=((ZZ00TA,$ROOT.00TA))
S M=((ZZ00TT,$ROOT.00TT))
S M=((ZZ00TU,$ROOT.00TU))
S M=((ZZ00TV,$ROOT.00TV))
S M=((ZZ00TW,$ROOT.00TW))
S M=((ZZ00TX,$ROOT.00TX))
```

INSTALLATION

19

INSTALLATION PROCESS

5

LOADING BATCH AND ON-LINE PROGRAMS

1

```

S M=((ZZ00TY,$ROOT.00TY))
S M=((ZZ00TZ,$ROOT.00TZ))
S M=((ZZ00T1,$ROOT.00T1))
S M=((ZZ00T2,$ROOT.00T2))
S M=((ZZ00T3,$ROOT.00T3))
S M=((ZZ00T4,$ROOT.00T4))
S M=((ZZ00T5,$ROOT.00T5))
S M=((ZZ00T6,$ROOT.00T6))
S M=((ZZ00T7,$ROOT.00T7))
S M=((ZZ00T8,$ROOT.00T8))
S M=((ZZ00T9,$ROOT.00T9))
S M=((ZZ00UD,$ROOT.00UD))
S M=((ZZ00ZZ,$ROOT.00ZZ))
S M=((ZZ0099,$ROOT.0099))
S M=((ZZCUAM,$ROOT.CUAM))
S M=((ZZCUEV,$ROOT.CUEV))
S M=((ZZCUMQ,$ROOT.CUMQ))
S M=((ZZCURQ,$ROOT.CURQ))
S M=((ZZCUSI,$ROOT.CUSI))

/*
//ET040 EXEC PGM=IMASPZAP
//SYSPRINT DD SYSOUT=$OUT
//SYSLIB DD DSN=$MODT,DISP=SHR
NAME $ROOT.00ZZ P200ZZ
VER 0005 D7F2
REP 0005 $HEXA
VER 0275 D7F2
REP 0275 $HEXA
VER 02AB D7F2
REP 02AB $HEXA
/*
===SEQ FOR SEC
//ET050 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3 DD UNIT=$UWK,SPACE=(TRK,20)
//IND1 DD DSN=PACD.MBR8,DISP=OLD,UNIT=$TAPE,
// VOL=(,RETAIN,SER=$TAPEI),LABEL=(04,SL)
//OUT1 DD DISP=OLD,DSN=----- <-- AUTHORIZED LIBRARY
COPY INDD=((IND1,R)),OUTDD=OUT1
S M=((PACSECRA,PACSECU8))
/*
//
===SEQ

```


INSTALLATION	PAGE	177
INSTALLATION PROCESS		19
ALLOCATION/LOADING OF SY PARAMETERS PDS		5
		2

19.5.2. ALLOCATION/LOADING OF SY PARAMETERS PDS

2. LOADING OF SYSTEM PARAMETERS PDS

Loading the PDS of System Parameters is made up of a JOB '\$PRFJ.SY' which contains the following steps:

ET010 : IEHPROGM : SCRATCH UNCATLG of PDS of SY parameters

ET020 : IEFBR14 : allocation of the parameters' PDS

ET030 : IEBUPDTE : loading of the parameters:

- DBD and PSB sources,
- Definitions (DELETE/DEFINE) of VSAM files,
- DELETE's and LISTCAT's of the journal file 'DJ',
- Verification (VERIFY) of VSAM files,
- REPRO and the load record of the SPA database 'DZ',
- REPRO and the load record of the DF work file,
- Definition of VSAM buffers (DFSVSAMP)
- PACCTRL member is for the Systems team only. It contains the description of the DSMS macro-instructions 'APPLCTN' and 'TRANSACT' which must be defined in the IMS Control Region.

(See JCL below.)

INSTALLATION

19

INSTALLATION PROCESS

5

ALLOCATION/LOADING OF SY PARAMETERS PDS

2

```

//SPRFJ.SY JOB ($CCPT), 'LOAD PARAM.', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//*          LOADING OF DSMS PARAMETERS IN 'SY' PDS          *
//*****
//*
//*  *-----*
//*  *          I M P O R T A N T          *
//*  *          -----*
//*  *          BEFORE EXECUTING THIS JOB, REPLACE ALL      *
//*  *          ':' BY '.' VIA THE EDITOR.                  *
//*  *-----*
//*
//*  *-----*
//*  *          N O T E          *
//*  *          -----*
//*  * THESE PARAMETERS CONTAIN THE 'SYSINS' FOR          *
//*  * ALLOCATING FILES AND DATABASES USED IN DSMS      *
//*  * MANAGEMENT FUNCTION. THE SIZES INDICATED CAN    *
//*  * BE ADAPTED TO YOUR NEEDS.                        *
//*  *-----*
//*
//*
//ET010 EXEC PGM=IEHPRGM
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD UNIT=$UNITP, VOL=SER=$VOLP, DISP=SHR
//SYSIN DD *
        UNCATLG DSN=$INDEXP..$ROOT.$ROOT.SY
        SCRATCH DSN=$INDEXP..$ROOT.$ROOT.SY, VOL=$UNITP=$VOLP
//*
//ET020 EXEC PGM=IEFBR14
//DDA DD DSN=$INDEXP..$ROOT.$ROOT.SY, DISP=(,CATLG,DELETE),
//      UNIT=$UNITP, VOL=SER=$VOLP,
//      DCB=(RECFM=FB, LRECL=80, BLKSIZE=6080),
//      SPACE=(TRK,(03,02,04))
//*
//ET030 EXEC PGM=IEBUPDTE, PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT1 DD DSN=$INDEXP..$ROOT.$ROOT.SY, DISP=SHR
//SYSUT2 DD DSN=$INDEXP..$ROOT.$ROOT.SY, DISP=SHR
//SYSIN DD *
:/ ADD NAME=PACDDA$SUF
    DBD NAME=PACDDA$SUF, ACCESS=(HIDAM, VSAM)
    DATASET DD1=PAC7DA$SUF, DEVICE=$UNITU, SIZE=4096
    SEGM NAME=PAC7DA, BYTES=(0350,0080)
    FIELD NAME=(CLEDA, SEQ, U), BYTES=40, START=03
    LCHILD NAME=(IPAC7D1, PACDD1$SUF), PTR=INDX
    DEBDGEN
    END
:/ ADD NAME=PACDD1$SUF
    DBD NAME=PACDD1$SUF, ACCESS=(INDEX, VSAM)
    DATASET DD1=PAC7D1$SUF, DEVICE=$UNITU
    SEGM NAME=IPAC7D1, BYTES=40
    FIELD NAME=(CLED1, SEQ, U), BYTES=40, START=01
    LCHILD NAME=(PAC7DA, PACDDA$SUF), INDEX=CLEDA
    DEBDGEN
    END
:/ ADD NAME=PACDDE$SUF
    DBD NAME=PACDDE$SUF, ACCESS=(HISAM, VSAM)
    DATASET DD1=PAC7DE$SUF, DEVICE=$UNITU,          $Y
            RECORD=98, SIZE=4096
    SEGM NAME=PAC7DE, BYTES=90
    FIELD NAME=(CLEDE, SEQ, U), BYTES=17, START=1
    DEBDGEN
    END
:/ ADD NAME=PACDDJ$SUF
    DBD NAME=PACDDJ$SUF, ACCESS=(HDAM, OSAM),          $Y
            RMNAME=(CGIPACR1, 20, 16000000)
    DATASET DD1=PAC7DJ$SUF,          $Y
            DEVICE=$UNITU, SIZE=4096
    SEGM NAME=PAC7DJ, BYTES=187
    FIELD NAME=(CLEDJ, SEQ, U), BYTES=7, START=1
    DEBDGEN

```

INSTALLATION

19

INSTALLATION PROCESS

5

ALLOCATION/LOADING OF SY PARAMETERS PDS

2

```

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

```

INSTALLATION

19

INSTALLATION PROCESS

5

ALLOCATION/LOADING OF SY PARAMETERS PDS

2

```

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

```

INSTALLATION

19

INSTALLATION PROCESS

5

ALLOCATION/LOADING OF SY PARAMETERS PDS

2

```

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

```

INSTALLATION

19

INSTALLATION PROCESS

5

ALLOCATION/LOADING OF SY PARAMETERS PDS

2

```

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

```

INSTALLATION

19

INSTALLATION PROCESS

5

ALLOCATION/LOADING OF SY PARAMETERS PDS

2

```

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

```


	PAGE	185
INSTALLATION		19
INSTALLATION PROCESS		5
LOADING OF BATCH OPERATION PROCEDURES		3

19.5.3. LOADING OF BATCH OPERATION PROCEDURES

3. LOADING OF BATCH OPERATION PROCEDURES

The load of batch operation procedures is made up of a JOB '\$PRFJ.PROC' which catalogues all batch operation procedures in the PROCLIB, via an IEBUPDTE. Each procedure is a member called \$radp.NNNN, where \$radp is the root chosen when generating the JCL, and NNNN the DSMS procedure standard name. The procedures are described in this manual.

(See JCL below.)

INSTALLATION
 INSTALLATION PROCESS
 LOADING OF BATCH OPERATION PROCEDURES

19
 5
 3

```
//SPRFJ.PROC JOB ($CCPT), 'LOADING PROCEDURES', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//ET010 EXEC PGM=IEBUPDTE, PARM=NEW
//*****
//*          CATALOGING THE DSMS PROCEDURES          *
//*****
//*
//*  *-----*
//*  *          I M P O R T A N T          *
//*  *          -----*
//*  *          BEFORE EXECUTING THIS JOB, REPLACE ALL *
//*  *          ':' BY '.' UNDER THE EDITOR.          *
//*  *-----*
//*
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2 DD DSN=$BIBP, DISP=SHR
//SYSIN DD DATA, DLM='%%'
:/ ADD NAME=$RADP.DARC
//* DSMS          : TRANSACTION ARCHIVAL          *
:/ ADD NAME=$RADP.DEXH
//* DSMS          : EXTRACTION OF TABLES FOR EXTERNAL LISTS *
:/ ADD NAME=$RADP.DEXP
//* DSMS          : EXTRACTION-UPDATE OF DSMS DATABASE      *
:/ ADD NAME=$RADP.DEXT
//* DSMS          : EXTRACTION OF BATCH TRANSACTIONS FOR DUPT *
:/ ADD NAME=$RADP.DINI
//* DSMS          : INITIALIZATION OF DSMS DATABASE          *
:/ ADD NAME=$RADP.DINS
//* DSMS          : LIST OF INSTALLED PROGRAMS              *
:/ ADD NAME=$RADP.DLDE
//* DSMS          : LOADING OF ERROR MESSAGES DATABASE 'DE' *
:/ ADD NAME=$RADP.DLDZ
//* DSMS          : LOADING SPA DATABASE 'DZ'              *
:/ ADD NAME=$RADP.DLVB
//* DSMS          : CHANGE LOW VALUE CHARACTERS INTO BLANKS *
:/ ADD NAME=$RADP.DPRT
//* DSMS          : PRINTING AND QUERY                      *
:/ ADD NAME=$RADP.DREN
//* DSMS          : RENAMING OF TABLE CODES, KEYWORDS AND SITES *
:/ ADD NAME=$RADP.DREO
//* DSMS          : DSMS REORGANIZATION                    *
:/ ADD NAME=$RADP.DRST
//* DSMS          : RELOADING-RESTORING OF THE DSMS DATABASE *
:/ ADD NAME=$RADP.DR8Q
//* DSMS          : 8.0.2 01 OR 02 DSMS DATABASE RETRIEVAL *
:/ ADD NAME=$RADP.DR80
//* DSMS          : RETRIEVAL FROM 8.0.1 DSMS DATABASE BACKUP *
:/ ADD NAME=$RADP.DSAV
//* DSMS          : BACKUP OF THE DSMS DATABASE            *
:/ ADD NAME=$RADP.DUPT
:/ ADD NAME=$RADP.DUPD
//* DSMS          : UPDATE OF THE DSMS DATABASE            *
:/ ADD NAME=$RADP.DXBJ
//* DSMS          : EXTRACTION OF DSMS JOURNAL              *
:/ ADD NAME=$RADP.DR15
//* DSMS          : RETRIEVAL FROM 1.2 OR 1.5 DATABASE BACKUP *
:/ ADD NAME=$RADP.DR5J
//* DSMS          : RETRIEVAL FROM 1.2 OR 1.5 DATABASE BACKUP *
:/ ADD NAME=$RADP.DEXQ
//* DSMS          : EXTRACTION-UPDATE OF DSMS DATABASE      *
:/ ADD NAME=$RADP.LDDF
//* DSMS          : LOADING DATABASE 'DF' (D.A.F.)          *
:/ ADD NAME=$RADP.DPDF
//* DSMS          : ACCESS FACILITY PRE-PROCESSING          *
```

	PAGE	187
INSTALLATION		19
INSTALLATION PROCESS		5
LOADING OF THE DBDLIB		4

19.5.4. LOADING OF THE DBDLIB

4. LOADING OF THE DBDLIB

The installation tape contains the databases DBDs as objects and sources.

JOB '\$PRFJ.DBD' loads the object DBDs in the DBDLIB, via IEBCOPY.

Loading the DBDLIB is to be executed only if the value cho- sen in JCL generation for:

- . the suffix of DBDs (parameter SUF) is 22,
- . the disk type of the databases (parameters UNITU and UNITV) is 3390,

and if the CI size (4,096) supplied in the parameters to DEFINE the databases has been maintained.

If this is not the case, the sources must be recompiled. Their contents must be checked, and particularly if the CI size has been changed, the SIZE and RMNANE macros must be adjusted. The sources have been catalogued in the Parameter file SY, under the name PACDxyy where yy has the value of the SUF parameter and xx the values of DA, D1, DE, DJ, DX, DZ, and F1.

Note: Use the IBM standard compilation procedure.

(See JCL below.)

INSTALLATION
INSTALLATION PROCESS
LOADING OF THE DBDLIB

19
5
4

```
//$PRFJ.DBD JOB ($CCPT), 'DBDLIB', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
//ET010 EXEC PGM=IEBCOPY  
//*****  
//*          LOADING OF DBDLIB          - DSMS *  
//*****  
//*  
//*-----*  
//* TO BE EXECUTED ONLY IF YOUR DEVICE TYPE IS '3390' AND IF *  
//* THE SUFFIX OF DBD IS '22'. OTHERWISE, YOU MUST COMPILE *  
//* THE DBD SOURCES FOUND IN THE PDS PARAMETERS FILE 'SY'. *  
//*-----*  
//*  
//SYSPRINT DD SYSOUT=$OUT  
//SYSUT3 DD UNIT=$UWK,SPACE=(TRK,5)  
//SYSUT4 DD UNIT=$UWK,SPACE=(TRK,5)  
//IND1 DD DSN=PACD.DBDLIB,DISP=OLD,UNIT=$UTAPE,LABEL=(06,SL),  
// VOL=(,RETAIN,SER=$TAPEI)  
//OUT1 DD DSN=$DBDLIB,DISP=SHR  
//SYSIN DD *  
COPY INDD=((IND1,R)),OUTDD=OUT1  
S M=PACDDA22  
S M=PACDDE22  
S M=PACDDJ22  
S M=PACDDX22  
S M=PACDDZ22  
S M=PACDDF22  
S M=PACDD122  
S M=PACDF122  
/*  
/*  
//
```

INSTALLATION	
INSTALLATION PROCESS	
LOADING OF THE PSBLIB	

19
5
5

19.5.5. LOADING OF THE PSBLIB

5. LOADING OF THE PSBLIB

The installation tape contains the PSBs, as objects and sources. JOB '\$PRFJ.PSB' loads the objects PCBs in the PSBLIB, via IEBCOPY. Loading the PSBLIB is to be executed only if the value chosen in the generation JCL for:

- . The suffix of DBDs (parameter SUF) is 22,
- . The root of the system (parameter ROOT) is P2,
- . The suffix of batch PSBs (parameter SUG) is 22.

If this is not the case, the sources must be recompiled. These sources have been catalogued in the Parameter Library SY, under the name XXXXXXzz for the BATCH PSB's where zz has the value of the SUG parameter, and rrYYYY for the ON-LINE PSB's where rr has the value of the ROOT parameter.

XXXXXX taking the values:

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

YYYY taking the values:

00ZZ and 0099

Note: Use the IBM standard compilation procedure.

(See JCL below.)

INSTALLATION
 INSTALLATION PROCESS
 LOADING OF THE PSBLIB

19
 5
 5

```
//SPRFJ.PSB JOB ($CCPT),'PSBLIB',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//ET010 EXEC PGM=IEBCOPY
//*****
//*                               LOADING OF PSBLIB                - DSMS *
//*****
//*
//*-----*
//* TO BE EXECUTED ONLY IF:                                         *
//* - YOUR CHOSEN ROOT SYSTEM IS ....: ROOT='P2'                   *
//* - THE SUFFIX OF THE DBD IS .....: SUF='22'                     *
//* - THE SUFFIX OF THE BATCH PSB IS : SUG='22'                     *
//* IF ONLY ONE OF THESE CONDITIONS IS NOT REALIZED, YOU           *
//* MUST COMPILE THE PDS SOURCES FOUND IN THE PDS PARAMETER        *
//* FILE 'SY'.                                                       *
//*-----*
//*
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3 DD UNIT=$UWK,SPACE=(TRK,15)
//SYSUT4 DD UNIT=$UWK,SPACE=(TRK,15)
//IND1 DD DSN=PACD.PSBLIB,DISP=OLD,UNIT=$UTAPE,LABEL=(07,SL),
// VOL=(,RETAIN,SER=$TAPEI)
//OUT1 DD DSN=$PSBLIB,DISP=SHR
//SYSIN DD *
COPY INDD=((IND1,R)),OUTDD=OUT1
S M=PDSB22
S M=PDSBAS22
S M=PDSEX22
S M=PDSINI22
S M=PDSR1022
S M=PDSR4022
S M=PDSUP022
S M=PDSV1022
S M=PDSV2022
S M=PDSXDT22
S M=PDSXTH22
S M=PDS30022
S M=PDS32022
S M=PDS38022
S M=PDS40022
S M=PDS45022
S M=PDS50022
S M=PDS60022
S M=PDS61022
S M=PDS70022
===SEQ FOR DAF
S M=PDSDAF22
===SEQ
S M=PTV09022
S M=PDSMS22
S M=P200ZZ
S M=P20099
/*
//*
//
```

INSTALLATION	PAGE	191
INSTALLATION PROCESS		19
INITIAL PREPARATION OF FILES		5
		6

19.5.6. INITIAL PREPARATION OF FILES

6. INITIAL PREPARATION OF FILES

This job is to be executed only when the system is installed for the FIRST time. It is called '\$PRFJ.PRE' and contains the following programs:

```
ET010 : IEHPROGM : SCRATCH UNCATLG of model DSCBs.  
ET020 : IEFBR14  : Allocation of DSCBs.  
ET030 : IDCAMS   : BLDG of data-group index & initializa-  
                : tion of BB file (DSMS Database backup)  
ET040 : IDCAMS   : BLDG of data-group index & initializa-  
                : tion of BJ file (archived Journal).  
ET050 : IDCAMS   : Load BB file  
ET060 : IEBGENER  : Load BJ file  
ET070 : $RADP.DLDZ : Load DZ SPA database  
ET075 : $RADP.LDDF : Load DF work file
```

Note: Use the IBM standard compilation procedure.

(See JCL below.)

INSTALLATION
 INSTALLATION PROCESS
 INITIAL PREPARATION OF FILES

19
 5
 6

```
//SPRFJ.PRE JOB ($CCPT), 'PREPARATION', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//*      JOB TO BE RUN ONLY WHEN FIRST INSTALLING DSMS      *
//*****
// JCLLIB ORDER=( $BIBP )
//ET010 EXEC PGM=IEHPRGM
//*****
//*      SCRATCH-UNCATLG OF DSCB      *
//*****
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD UNIT=$UNITP, VOL=SER=$VOLP, DISP=SHR
//SYSIN DD *
    UNCATLG DSN=$INDEXQ..DSCB.$ROOT.$FILE.BB
    SCRATCH DSN=$INDEXQ..DSCB.$ROOT.$FILE.BB, VOL=$UNITP=$VOLP
    UNCATLG DSN=$INDEXQ..DSCB.$ROOT.$FILE.BJ
    SCRATCH DSN=$INDEXQ..DSCB.$ROOT.$FILE.BJ, VOL=$UNITP=$VOLP
/*
//ET020 EXEC PGM=IEFBR14
//DDA DD DSN=$INDEXQ..DSCB.$ROOT.$FILE.BB, DISP=( ,CATLG,DELETE),
//     SPACE=(TRK,(0)), VOL=SER=$VOLP, UNIT=$UNITP,
//     DCB=(RECFM=VB, LRECL=354, BLKSIZE=6376)
//DDB DD DSN=$INDEXQ..DSCB.$ROOT.$FILE.BJ, DISP=( ,CATLG,DELETE),
//     SPACE=(TRK,(0)), VOL=SER=$VOLP, UNIT=$UNITP,
//     DCB=(RECFM=FB, LRECL=180, BLKSIZE=6300)
//*
//*****
//*      BUILDING GENERATION FILES' INDEX      *
//*****
//ET030 EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$CATV, DISP=SHR
//*: DD DSN=$CATU, DISP=SHR
//DD1 DD DSN=$INDEXQ..$ROOT.$FILE.BB, DISP=( ,CATLG,DELETE),
//     UNIT=$UNITO, VOL=SER=$VOLO, SPACE=(TRK,0),
//     DCB=$INDEXQ..DSCB.$ROOT.$FILE.BB
//SYSIN DD *
    DEFINE GENERATIONDATAGROUP -
        (NAME ($INDEXQ..$ROOT.$FILE.BB) LIMIT (3) SCR)
//SYSPRINT DD SYSOUT=$OUT
//*
//ET040 EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$CATV, DISP=SHR
//*: DD DSN=$CATU, DISP=SHR
//DD1 DD DSN=$INDEXQ..$ROOT.$FILE.BJ, DISP=( ,CATLG,DELETE),
//     UNIT=$UNITO, VOL=SER=$VOLO, SPACE=(TRK,0),
//     DCB=$INDEXQ..DSCB.$ROOT.$FILE.BJ
//SYSIN DD *
    DEFINE GENERATIONDATAGROUP -
        (NAME ($INDEXQ..$ROOT.$FILE.BJ) LIMIT (3) SCR)
//SYSPRINT DD SYSOUT=$OUT
//*
//ET050 EXEC PGM=IDCAMS
//*****
//*      LOADING OF THE BB FILE      *
//*****
//*:STEPCAT DD DSN=$CATV, DISP=SHR
//*: DD DSN=$CATU, DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DD1 DD DSN=PAC.BB, DISP=OLD, UNIT=$UTAPE, LABEL=(08,SL),
//     VOL=( ,RETAIN, SER=$TAPEI)
//DD2 DD DSN=$INDEXQ..$ROOT.$FILE.BB(+1), DISP=( ,CATLG,DELETE),
//     VOL=SER=$VOLO, SPACE=(TRK,(100,10),RLSE), UNIT=$UNITO,
//     DCB=$INDEXQ..DSCB.$ROOT.$FILE.BB
//SYSIN DD *
    REPRO INFILE(DD1) OUTFILE(DD2)
//ET060 EXEC PGM=IEBGENER
//*****
//*      LOADING OF THE BJ FILE      *
//*****
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT1 DD DUMMY, DCB=$INDEXQ..DSCB.$ROOT.$FILE.BJ
//SYSUT2 DD DSN=$INDEXQ..$ROOT.$FILE.BJ(+1), DISP=( ,CATLG,DELETE),
```


INSTALLATION
INSTALLATION PROCESS
INITIAL PREPARATION OF FILES

19
5
6

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

	PAGE	194
INSTALLATION		19
INSTALLATION PROCESS		5
LOADING OF THE ERROR MESSAGE FILE		7

19.5.7. LOADING OF THE ERROR MESSAGE FILE

7. LOADING OF DSMS ERROR MESSAGE FILE

Job '\$PRFJ.DE' creates the system database (DE) of error messages. It includes the following program:

ET010 : \$RADP.DLDE : Load of DE file using the sequential
file provided on the tape.

(See JCL below.)

INSTALLATION
INSTALLATION PROCESS
LOADING OF THE ERROR MESSAGE FILE

19
5
7

```
//$PRFJ.DE JOB ($CCPT),'LOADING -DE-',CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
//*****  
//*          LOADING OF DSMS ERROR MESSAGE DATABASE          - DSMS *  
//*****  
// JCLLIB ORDER=( $BIBP )  
//*  
//ET010 EXEC $RADP.DLDE  
//PDSV10.PAC7IN DD DSN=PAC.DE,DISP=OLD,UNIT=$UTAPE,LABEL=(09,SL),  
//          VOL=( ,RETAIN,SER=$TAPEI )  
//*  
//
```

INSTALLATION	
INSTALLATION PROCESS	
COMPILATION OF ACBs	

19
5
8

19.5.8. COMPILATION OF ACBs

8. COMPILATION OF ACBs

This JOB creates all ACBs allowing DSMS on-line use.

Note: Use the IBM standard compilation procedure.

IMPORTANT:

The DBDLIB must be concatenated with the VA Pac system's DBDLIB. (See parameter \$DBDLIO.)

(See JCL below.)

INSTALLATION
INSTALLATION PROCESS
COMPILATION OF ACBs

PAGE

197

19
5
8

```
//$PRFJ.ACB JOB ($CCPT), 'ACBGEN', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//ET010 EXEC ACBGEN, SOUT=$OUT, PSB='$PSBLIB',
// DBD='$DBDLIB', ACB='$ACBLIB'
//*****
//* COMPILATION OF ACBs - DSMS *
//*****
//*
//*-----*
//* THIS JOB MUST BE EXECUTED AFTER LOADING DBDLIB AND *
//* PSBLIB (EITHER BY LOADING OF OBJECT MODULES OR BY *
//* COMPILATION OF DBD AND PSB SOURCES). *
//* *
//* IMPORTANT: THE DSMS DBDLIB MUST BE CONCATENED WITH THE *
//* VA PAC DBDLIB. *
//*-----*
//*
//G.SYSIN DD *
BUILD DBD=( PACDDA$SUF, PACDDC$SUF, PACDDE$SUF, PACDDJ$SUF)
BUILD DBD=( PACDDX$SUF, PACDDZ$SUF, PACDDF$SUF, PACDD1$SUF)
BUILD DBD=( PACDD3$SUF, PACDF1$SUF)
BUILD PSB=( $ROOT.00ZZ, $ROOT.0099)
/*
/*
/*
```

INSTALLATION	PAGE	198
INSTALLATION PROCESS		19
COMPLEMENT: ENTRY-POINT SOURCES FOR USER CHECKS		5
		9

19.5.9. COMPLEMENT: ENTRY-POINT SOURCES FOR USER CHECKS

9. COMPLEMENT: ENTRY-POINT SOURCES FOR USER CHECKS

This complementary installation should only be performed by users who wish to add checks on Change, Event, Query, Report or Site definition screens and by DAF users.

Job '\$prfj.7SRC'

STEP1: IDCAMS : PDS deletion,

STEP2: IEFBR14 : PDS allocation,

STEP3: IEBCOPY : PDS members loading.

9.1 INSTALLATION OF SOURCES FOR USER CHECKS

Five batch sub-programs and 5 on-line sub-programs are shipped for the DSMS administrator to insert additional controls in them. They must be compiled and linked in the DSMS load-module libraries.

To make them active, the activation must be specified during restoration. (See the chapter dedicated to the DRST procedure.)

INSTALLATION OF BATCH TRANSACTIONS FOR DAF

These transactions can be used for writing programs calling the DAF facility. They must therefore be updated in the VA Pac database via the UPDT procedure. (See next Subchapter 'Installation of the DAF Environment').

INSTALLATION
 INSTALLATION PROCESS
 COMPLEMENT: ENTRY-POINT SOURCES FOR USER CHECKS

19
 5
 9

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

	PAGE	200
INSTALLATION		19
INSTALLATION PROCESS		5
COMPLEMENT: ENTRY-POINT SOURCES FOR USER CHECKS		9

10.2. INSTALLATION OF THE DAF ENVIRONMENT

The DAF facility allows the translation of SQL queries, written in user programs, for access to the DSMS Database, through the generation of data and VA Pac access sub-program calls in the COBOL source generated from these programs.

The pre-processor processes the generated programs in order to perform this translation. It includes the DAFP10 program installed in the batch load-module library MBR8.

To process the generated programs that use DAF, the DPDF procedure is available. It should be used in one of the following ways:

- . Request this procedure in the Optional Control Cards in front of/in back of program, which are combined with the link-edit compilation JCL.
- . Call this procedure after the execution of the standard GPRT procedure, from which the generated flow will be retrieved.
- . Use any other method best suited with the characteristics of the site.

(Refer to the subchapter dedicated to the DPDF procedure.)

ONE DAF SUB-PROGRAM is provided in the installation deck:

PDTBDF for DAF standard requests

NOTE:

This sub-program should be transferred in the user program library(ies), either to be included in the "Link-edit" of the user programs (static call), either to be called for execution (dynamic call).

A JCL EXAMPLE of user batch program calling DAF is provided in the JCL PDS, in the PACSDAF member (see example below).

The work file necessary for the operation of DAF in on-line mode has a CICS-imposed DDNAME of the \$ROOT.\$ROOT.DF format. This DDNAME must be the only one for all programs accessing the same DSMS Database.

Data Element, Data Structure and Segment entities used to write programs involving DAF, are provided as batch transactions in the DAFDIC member of the SRC Complements' PDS.

IMPORTANT:

Loading the 'DAF dictionary' in the VA Pac database via the UPDT batch update procedure is the responsibility of the Database Administrator, who must make sure that the codes of the entities provided do not conflict with entities that are already defined in the Database.

In order to avoid compatibility conflicts between the site's Dictionary and entities provided for the DAF facility, it is recommended to create an independent library network that will be accessed by the site's DAF utilities. However, this Dictionary may be loaded in the same library as the PAF Dictionary.

INSTALLATION

19

INSTALLATION PROCESS

5

COMPLEMENT: ENTRY-POINT SOURCES FOR USER CHECKS

9

```

//*****
//* DSMS 2.5
//*          - JCL EXAMPLE -
//*          EXECUTION OF A USER D.A.F. BATCH PROGRAM
//*****
//DAFBATCH  PROC FILE=$FILE,          PHYSICAL-DATABASE NUMBER
//          ROOT=$ROOT,              ROOT OF DSMS SYSTEM
//          INDEX='$INDEX',          VSAM INDEX
//*:        SYSTCAT='$CATV',          VSAM SYSTEM CATALOG
//*:        VSAMCAT='$CATU',          VSAM USER CATALOG
//          OUT='$OUT',              OUTPUT CLASS
//          INDEXP='$INDEXP',        NON-VSAM FILE INDEX
//          STEPLIB='$MODB',          LIBRARY OF BATCH LOAD-MODULES
//          PSBLIB='$PSBLIB',        LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',        LIBRARY OF DBD'S
//          DBDLIO='$DBDLIO',        LIBRARY OF VA PAC DBD'S
//          RESLIB='$RESLIB',        IMS RESLIB
//          PROCLIB='$PRCLIB',       IMS PROCLIB
//          UWK=$UWK,                WORK UNIT
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//WITHDAF  EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,_____,_____$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//          COND=(00,NE,EXISDJ)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//          DD DSN=&DBDLIO,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7DA   DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1   DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DC   DD DSN=&INDEX..&ROOT.&FILE.DC,DISP=SHR
//PAC7D3   DD DSN=&INDEX..&ROOT.&FILE.D3,DISP=SHR
//PAC7DX   DD DSN=&INDEX..&ROOT.&FILE.DX,DISP=SHR
//PAC7DE   DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PAC7DF   DD DSN=&INDEX..&ROOT.&ROOT.DF,DISP=SHR
//PAC7F1   DD DSN=&INDEX..&ROOT.&ROOT.F1,DISP=SHR
//----- DD DSN=---
//----- DD DSN=---
//----- DD DSN=---
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//          PEND
//DAFBATCH EXEC DAFBATCH

```

	PAGE	203
INSTALLATION		19
INSTALLATION PROCESS		5
LOADING OF THE DSMS TEST DATABASE		10

19.5.10. LOADING OF THE DSMS TEST DATABASE

10. LOADING OF THE DSMS TEST DATABASE

Before running the tests, the DSMS test database must be reloaded through the execution of the '\$PRFJ.DS' JOB.

'\$PRFJ.DS' executes the DRST procedure using as input the backup previously loaded on disk by the '\$PRFJ.PRE' job (Step ET050 in 6 above).

(See JCL below.)

INSTALLATION

19

INSTALLATION PROCESS

5

LOADING OF THE DSMS TEST DATABASE

10

```

//$PRFJ.DS JOB ($CCPT),'TESTS',CLASS=$CLASSJ,
//  MSGCLASS=$MSGCL
// *
//*****
//*          LOADING OF DSMS TEST FILES          *
//*****
// *
//*-----*
//* AFTER A SUCCESSFUL EXECUTION OF THE 'DRST' PROCEDURE, THE *
//* DSMS DATABASE IS INSTALLED. THE INITIAL CONNECTION TO THE *
//* DSMS DATABASE IS EXECUTED AS FOLLOWS:                *
//*   - ACCESS THE DSMS DATABASE                          *
//*   - IN THE SIGN-ON SCREEN, ENTER,                     *
//*     THE USER CODE.....: '*****'                    *
//*     THE PASSWORD.....: '*****'                      *
//*     THE CHOICE.....: 'HT'                             *
//* (FOR THE AUTHORIZED CHOICES IN THIS CASE, CONSULT THE *
//* 'DINI' PROCEDURE IN THE OPERATIONS MANUAL).          *
//*-----*
//  JCLLIB ORDER=( $BIBP)
//STEP01 EXEC $RADP.DRST
//  RE
// *
//

```

19.6. LIST OF INSTALLED PROGRAMS

11. LIST OF INSTALLED PROGRAMS

This list can be obtained via the '\$PRFJ.INSL' job through the execution of the '\$RADP.DINS' procedure. It contains:

- . The list of Batch and On-line programs with compilation dates.

It is recommended to keep this list in order to transmit installation references to IBM in case of system malfunctioning.

(See JCL below.)

INSTALLATION
LIST OF INSTALLED PROGRAMS

PAGE

206

19
6

```
//$PRFJ.INSL JOB ($CCPT), 'INSTALLATION LIST', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// *  
//*****  
// * THE FOLLOWING JOB PROVIDES LISTS OF INSTALLED PROGRAMS *  
//*****  
// *  
// JCLLIB ORDER=( $BIBP )  
//STEP01 EXEC $RADP.DINS  
$ROOT  
// *  
//
```

INSTALLATION
LIST OF INSTALLED PROGRAMS

19
6

```

//*****
//* DSMS 2.5
//*
//* LIST OF INSTALLED PROGRAMS
//*****
//$RADP.DINS PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, VA PAC-DSMS SYSTEM ROOT
// INDEXP='$INDEXP', INDEX OF NON-VSAM FILES
// INDEX='$INDEX', INDEX OF VSAM FILES
//*: VSAMCAT='$CATU', VSAM USER CATALOG
// OUT='$OUT', OUTPUT CLASS
// MODB='$MODB', LIBRARY OF BATCH LM
// MODT='$MODT', LIBRARY OF ON-LINE LM
// PSBLIB='$PSBLIB', IMS PSBLIB
// DBDLIB='$DBDLIB', IMS DBDLIB
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS WORKLIB
// UWK=$UWK, WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//* INPUT :
//* - 1ST LINE: ROOT OF THE DSMS SYSTEM ..(COL.3 LENGTH =2)
//* - OTHER LINES IF SELECTION OF PROGRAMS IS NEEDED :
//* ONE LINE PER PROGRAM: PROGRAM CODE .....(COL.3 LENGTH=6)
//*****
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&MODB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&PAC7MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDDA DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//DDD1 DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//SYSIN DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDA),DISP=SHR
// DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFD1),DISP=SHR
//*
//PDSXDT EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,PDSXDT,PDSXDT$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&MODB,DISP=SHR
// DD DSN=&MODT,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7DA$SUF DD DSN=&INDEX..&ROOT.&FILE.DA,DISP=SHR
//PAC7D1$SUF DD DSN=&INDEX..&ROOT.&FILE.D1,DISP=SHR
//PAC7DE$SUF DD DSN=&INDEX..&ROOT.&ROOT.DE,DISP=SHR
//PACDDS DD SYSOUT=&OUT
//PACDMB DD DSN=&&PAC7MB,DISP=(OLD,PASS)
//*

```

19.7. USE TESTS

USE TESTS

DSMS INSTALLATION TESTS

These tests include three steps:

- . On-line use tests,
- . Extraction utility test,
- . Database management tests.

1. ON-LINE USE TESTS

Open the test Database files in on-line mode. Log in with the user 'TEST'. Use password 'IBM'. Perform screen branchings and updates.

2. EXTRACTION TEST

Run '\$prfjDEXT' (DEXT procedure). This job extracts elements from the test Database.

For this test, the Database files can remain open in on-line mode.

3. DATABASE MANAGEMENT TESTS

The purpose of these tests is to execute Database management procedures.

The following steps must be performed in the indicated order:

- . Archiving the journal created during the use tests: Run '\$prfjDARC' job, creates a BJ(1) file.
- . Direct backup of the Database: Run '\$prfjDSAV' job, creates a BB(1) file.
- . Database restoration from BJ(1) archive and BB(1) Database backup: Run '\$prfjDRST' job.

During all these tests, the Database files must be closed to on-line access.

After the Database is restored, open the Database files and perform another set of quick operational tests on-line

INSTALLATION
USE TESTS
TEST JCL: DEXT

PAGE

210

19
7
1

19.7.1. TEST JCL: DEXT

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

INSTALLATION
USE TESTS
TEST JCL: DARC

19
7
2

19.7.2. TEST JCL: DARC

```
//$PRFJ.DAR JOB ($CCPT), 'DARC', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// JCLLIB ORDER=( $BIBP )  
//*****  
//*          TESTING THE JOURNAL ARCHIVAL PROCEDURE          *  
//*****  
//DARC EXEC $RADP.DARC
```

INSTALLATION
USE TESTS
TEST JCL: DSAV

19
7
3

19.7.3. TEST JCL: DSAV

```
//$PRFJ.DSA JOB ($CCPT), 'DSAV', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// JCLLIB ORDER=( $BIBP )  
//*****  
//*          TESTING THE DATABASE BACKUP          *  
//*****  
//DSAV EXEC $RADP.DSAV
```

INSTALLATION
USE TESTS
TEST JCL: DRST

19
7
4

19.7.4. TEST JCL: DRST

```
//$PRFJ.DRS JOB ($CCPT), 'DRST', CLASS=$CLASSJ,  
// MSGCLASS=$MSGCL  
// JCLLIB ORDER=( $BIBP )  
//*****  
//*          TESTING THE RESTORATION WITH THE JOURNAL          *  
//*          *  
//* AFTER THE RESTORATION, MAKE A FEW QUICK TESTS FOR ON-LINE *  
//* FUNCTIONNING, AFTER HAVING RE-OPENED THE DATABASE FILES. *  
//*****  
//DRST EXEC $RADP.DRST  
RE REC  
//*
```

INSTALLATION
USE TESTS
TEST JCL: DXBJ

PAGE

214

19
7
5

19.7.5. TEST JCL: DXBJ

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

19.8. SYSTEM REINSTALLATION

RE-INSTALLATION OF A DSMS VERSION

DSMS must be re-installed when a new version of the software comes out following corrections and enhancements.

This new version, identified by a number is made of:

A sub-release, identified by a number is composed of:

- . the product installation cartridge (or tape),
- . the list of the corrected abends,
- . additional instructions that might be included to complete the reinstallation steps described in this Subchapter.

In general, only system files and program libraries are affected by a new release.

Three cases are possible:

Case 1: installation JCLs have been kept

Case 2: installation JCLs must be re-generated:
Standard reinstallation.

Case 3: installation JCLs must be re-generated:
Non standard reinstallation.

CASE 1: INSTALLATION JCLs HAVE BEEN KEPT

For a STANDARD REINSTALLATION, run the jobs contained in the following JCLs:

- PACSLOAD: \$prfj.LOAD program initialization.
- PACSDE : \$prfj.DE error message initialization.

1) PACSLOAD: Batch and On-Line load modules re-initialization (modify the name of the tape in in the VOL=SER= parameter).

NOTE: the provided job includes: load-modules libraries deletion, their allocation and all programs copy. Two procedures can be performed:

- . Complete job execution: in this case, if the libraries contain programs not coming from the installation tape (user programs) or adapted programs, save them before running the job;
- . Execution of the programs copy step only (IEBCOPY): (in this case, delete the programs first in order to avoid library space problems).

2) PACSDE: IDCAMS DELETE/DEFINE and REPRO of the error messages and documentation DE file. (Modify the tape name in VOL=SER= parameter).

CASE 2: YOU MUST RE-GENERATE INSTALLATION JCLs FOR A
STANDARD REINSTALLATION

To obtain more details about the procedures to perform, refer to Subchapters 'Initial JCL' and 'Complete JCL Installation'.

To re-generate JCLs, run MM1JCL utility again with the parameters set for the site installation and the JCLs needed for the reinstallation.

Add lines in the SYSIN in order to select the following JCLs modules:

===SELM PACSLOAD

===SELM PACSDE

Check resulting JCLs. Perform the reinstallation according to the steps of CASE 1.

CASE 3: YOU MUST RE-GENERATE INSTALLATION JCLs FOR A NON-
STANDARD REINSTALLATION

To get the JCLs, see CASE 2.

Once you have the JCLs, follow the special instructions indicated in the notice provided with the sub-release.

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
RETRIEVAL OF DSMS 8.0.1 --> DSMS 2.5

(DR80)

PAGE 218

20

20. RETRIEVAL OF DSMS 8.0.1 --> DSMS 2.5 (DR80)

20.1. OPERATIONS TO CARRY OUT

OPERATIONS TO CARRY OUT

The retrieval of a DSMS 8.0 (or 8.0.1) database and its adaptation to the new release requires the following operations:

- . 8.0/8.0.1 archival of the Database (DARC procedure).
- . 8.0/8.0.1 backup of the Database, producing a 8.0/8.0.1 file called BB (DSAV procedure).

Then, with the NEW INSTALLATION, execute the following procedures:

- . Convert the 8.0/8.0.1 (BB) DSMS database backup to the new format (DR80 procedure).
- . Reorganize the backup in order to rebuild the DX cross-references in the new format (DREO procedure).
- . Restore the database (DRST procedure).

20.2. USER INPUT

USER INPUT

User input allows product codes to be changed from one character to three. It is composed of 1 to n lines with each line sub-divided into groups of four characters starting from column one.

```
+-----+-----+-----+-----+
! POS.! LEN.! VALUE ! MEANING
+-----+-----+-----+-----+
! 1 ! 1 ! 'P' ! Old product code
! 2 ! 3 ! 'PRO' ! New product code
! ! ! ! Each group of 4 characters can be
! ! ! ! repeated a maximum of 20 times per
! ! ! ! line.
+-----+-----+-----+-----+
```

20.3. BACKUP RETRIEVAL

DR80: DESCRIPTION OF STEPS

INPUT RECOGNITION: PTU001

CONVERSION PREPARATION: PDSR8B

.Permanent input file
-Backup of DSMS 8.0.1 DSMS database
PACDBB : DSN=&OLDBB

.Output work files:
PACDIQ : DSN=&&PACDIQ
PACDIT : DSN=&&PACDIT
PACDIW : DSN=&&PACDIW
PACDLA : DSN=&&PACDLA

.Sort files:
SORTWK01, SORTWK02, SORTWK03

.Input transaction file:
PACDMB : DSN=&&DR80MB

.Output report:
-Retrieval report
PACDIK

8.0.2 BACKUP CONVERSION: PDSR8C

.Input work files:
PACDIQ : DSN=&&PACDIQ
PACDIT : DSN=&&PACDIT
PACDIW : DSN=&&PACDIW
PACDLA : DSN=&&PACDLA

.Permanent input file:
-Backup of DSMS 8.0.1 database
PACDBB : DSN=&OLDBB

.Permanent output file:
-Backup of converted DSMS database
PACDB1 : DSN=&BBOLD

RETRIEVAL OF DSMS 8.0.1 --> DSMS 2.5
BACKUP RETRIEVAL

(DR80)

PAGE

222

20
3

.Input transaction file:
PACDMB : DSN=&&DR80MB

.Sort files:
SORTWK01, SORTWK02, SORTWK03

.Sort files

.Output report:
-Retrieval report
PACDIO

CONVERSION OF BACKUP 1.2 TO 2.5 FORMAT: PDSR15

.Permanent input file:
-Backup file of DSMS 1.2 or 1.5 Database
PACDBB : DSN=&BBOLD

.Permanent output file:
-Backup of converted DSMS Database
PACDB1 : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

.Output report:
-Printing report
PACDIK

20.4. EXECUTION JCL

```

//*****
//* DSMS 2.5
//* - RETRIEVAL OF DSMS DATABASE 8.0.1 -
//*****
//$RADP.DR80 PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, DSMS SYSTEM ROOT
// OUT='$OUT', OUTPUT CLASS
// INDEXQ='$INDEXQ', GENERATION-FILE INDEX
// CYL=(4,1)', SORT WORKS SIZE
// SPABB='CYL,(10,5)', SPACE OF BACKUP
// SPAWK='CYL,(10,5)', SPACE OF WORK FILES
// UWK=$UWK, WORK UNIT
// OLDBB=, DSN OF DSMS DATABASE 8.0.1 BACKUP
// VOLS='SER=$VOLO', GENERATION FILE VOLUME
// UNITS='$UNITS', GENERATION FILE UNIT
// STEPLIB='$MODB', LIBRARY OF BATCH LM
// SORTLIB='$BIBT' SORT LIBRARY
//*-----*
//*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&DR80MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//PDSR8B EXEC PGM=PDSR8B
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDBB DD DSN=&OLDBB,DISP=OLD
//PACDMB DD DSN=&&DR80MB,DISP=(OLD,PASS)
//PACDLA DD DSN=&&PACDLA,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=1,BLKSIZE=1000)
//PACDIQ DD DSN=&&PACDIQ,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=284,BLKSIZE=5680)
//PACDIT DD DSN=&&PACDIT,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=104,BLKSIZE=2080)
//PACDIW DD DSN=&&PACDIW,DISP=(,PASS),
// UNIT=&UWK,SPACE=(&SPAWK,RLSE),
// DCB=(RECFM=FB,LRECL=92,BLKSIZE=2024)
//PACDIK DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PDSR8C EXEC PGM=PDSR8C
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PACDBB DD DSN=&OLDBB,DISP=OLD
//PACDMB DD DSN=&&DR80MB,DISP=(OLD,PASS)
//PACDLA DD DSN=&&PACDLA,DISP=(OLD,PASS)
//PACDIQ DD DSN=&&PACDIQ,DISP=(OLD,PASS)
//PACDIT DD DSN=&&PACDIT,DISP=(OLD,PASS)
//PACDIW DD DSN=&&PACDIW,DISP=(OLD,PASS)
//PACDB1 DD DSN=&&BBOLD,DISP=(,PASS),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=(&SPABB,RLSE),
// DCB=(RECFM=VB,BLKSIZE=6376,LRECL=354)

```

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


VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
RETRIEVAL OF DSMS 8.0.2 01/02 --> DSMS 2.5 (DR8Q)

PAGE 225

21

21. RETRIEVAL OF DSMS 8.0.2 01/02 --> DSMS 2.5 (DR8Q)

21.1. OPERATIONS TO CARRY OUT

OPERATIONS TO CARRY OUT

NOTE: this chapter relates to databases already installed in 8.0.2 (version 01 or 02); if a DR8X or DR80 retrieval procedure was executed, do not perform this new retrieval.

Installing the new DSMS release requires retrieving the DSMS database queries, which includes the following steps:

Using the 8.0.2 01 or 02 procedures:

1. DSMS database archive (DARC)
2. DSMS database backup (DSAV)

Then, with the NEW INSTALLATION, execute the following procedures:

3. Retrieval of the BB backup file (DR8Q).
4. DSMS reorganization (DREO).
5. DSMS database restoration (DRST).

EXECUTION CONDITIONS

None.

However, to ensure the consistency of the retrieved database, it is recommended to close the database to on-line use.

USER INPUT

None.

21.2. BACKUP RETRIEVAL

DR8Q: DESCRIPTION OF STEPS

QUERY RETRIEVAL: PDSR8Q

.Permanent input file:
-Backup of DSMS 8.0.2 01/02 Database
PACDBB : DSN=&OLDBB

.Output work file:
PACDIQ : DSN=&&PACDIQ

.Sort files:
SORTWK01, SORTWK02, SORTWK03

MERGE: PDSR8R

.Input work file
PACDIQ : DSN=&&PACDIQ

.Permanent input file:
-Backup of DSMS Database 8.0.2 01/02
PACDBB : DSN=&OLDBB

.Output Permanent file :
-Backup of retrieved DSMS Database
PACDB1 : DSN=&BBOLD

CONVERSION OF BACKUP 1.2 TO 2.5 FORMAT: PDSR15

.Permanent input file:
-Backup file of DSMS 1.2 or 1.5 Database
PACDBB : DSN=&BBOLD

.Permanent output file:
-Backup of converted DSMS Database
PACDB1 : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

.Output report:
-Printing report
PACDIK

21.3. EXECUTION JCL

```
//*****  
/* DSMS 2.5 *  
/* - 8.0.2 01 / 02 DSMS DATABASE RETRIEVAL INTO 2.5 - *  
//*****  
//$RADP.DR8Q PROC ROOT='$ROOT',  
// FILE='$FILE',  
// INDEXQ='$INDEXQ', INDEX OF USER NON-VSAM FILES  
// STEPLIB='$MODB', LIBRARY OF BATHC LM  
// OUT='$OUT', OUTPUT CLASS  
// SORTLIB='$BIBT', SORT LIBRARY  
// CYL=3, SORTWORK SPACE  
// UWK=$UWK, WORK UNIT  
// OLDBB=, 8.0.2 02 DSMS BACKUP DSNAME  
// VOLS='SER=$VOLO', VOLUME OF BACKUP (BB)  
// UNITS='$UNITO', BACKUP UNIT (DISK OR TAPE)  
// SPAWK='(CYL,(20,2))', WORK FILE SPACE  
// SPABB='(TRK,(45,5),RLSE)' SPACE OF BACKUP (IF DISK)  
//*****  
//PDSR8Q EXEC PGM=PDSR8Q,REGION=4096K  
//*****  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//SORTLIB DD DSN=&SORTLIB,DISP=SHR  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//PACDBB DD DSN=&OLDBB,DISP=OLD  
//PACDIQ DD DSN=&&PACDIQ,DISP=(,PASS),  
// UNIT=&UWK,SPACE=&SPAWK,  
// DCB=(RECFM=FB,LRECL=284,BLKSIZE=5680)  
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)  
//SYSUDUMP DD SYSOUT=&OUT  
//PDSR8R EXEC PGM=PDSR8R,REGION=4096K  
//*****  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//SYSOUT DD SYSOUT=&OUT  
//PACDBB DD DSN=&OLDBB,DISP=OLD  
//PACDB1 DD DSN=&&BBOLD,  
// DISP=(,PASS),  
// UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,  
// DCB=(RECFM=VB,BLKSIZE=6376,LRECL=354)  
//PACDIQ DD DSN=&&PACDIQ,DISP=(OLD,PASS)  
//SYSUDUMP DD SYSOUT=&OUT  
//*  
//PDSR15 EXEC PGM=PDSR15  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//PACDBB DD DSN=&&BBOLD,DISP=(OLD,PASS)  
//PACDB1 DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),  
// UNIT=&UNITS,VOL=&VOLS,  
// SPACE=&SPABB,  
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB  
//PACDIK DD SYSOUT=&OUT  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//SYSRINT DD SYSOUT=&OUT  
//SYSUDUMP DD SYSOUT=&OUT  
//*
```

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
RETRIEVAL OF DSMS 1.2 OR 1.5 --> DSMS 2.5

PAGE 229

22

22. RETRIEVAL OF DSMS 1.2 OR 1.5 --> DSMS 2.5

22.1. OPERATIONS TO CARRY OUT

OPERATIONS TO CARRY OUT

The retrieval of a DSMS 1.2 (or 1.5) database and its adaptation to the new release requires the following operations:

- . 1.2/1.5 archival of the Database (DARC procedure).
- . 1.2/1.5 backup of the Database, producing a 1.2/1.5 file called BB (DSAV procedure).

Then, with the NEW INSTALLATION, execute the following procedures:

- . Convert the 1.2/1.5 (BB) DSMS database backup to the new format (DR15 procedure).
- . Reorganize the backup in order to rebuild the DX cross-references (DX file) in the new version format (DREO procedure).
- . Restore the database (DRST procedure).

NOTE:

It is possible to retrieve the sequential version of the journal (Rel. 1.2 or 1.5) with the DR5J procedure.

22.2. 'DR15' PROCEDURE - DESCRIPTION OF STEPS

DR15: DESCRIPTION OF STEPS

RETRIEVAL OF DSMS 1.2 / 1.5: PDSR15

.Permanent input file:

-DSMS 1.2 or 1.5 database backup
PACDBB : DSN=&OLDBB

.Permanent output file:

-Retrieved DSMS database backup
PACDB1 : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

.Output report:

-Printing report
PACDIK

22.3. 'DR15' PROCEDURE - EXECUTION JCL

```
//*****  
//* DSMS 2.5 *  
//* - RETRIEVAL OF DSMS DATABASE 1.2 / 1.5 INTO 2.5 - *  
//* *  
//*****  
//$RADP.DR15 PROC FILE=$FILE, PHYSICAL-DATABASE NUMBER  
// ROOT=$ROOT, DSMS SYSTEM ROOT  
// OUT='$OUT', OUTPUT CLASS  
// INDEXQ='$INDEXQ', GENERATION-FILE INDEX  
// SPABB='(TRK,(45,5),RLSE)', SPACE OF BACKUP  
// OLDBB=, DSN 1.2 -1.5 DATABASE BACKUP  
// VOLS='SER=$VOLO', GENERATION FILE VOLUME  
// UNITS='$UNITO', GENERATION FILE UNIT  
// STEPLIB='$MODB' LIBRARY OF BATCH LM  
//*-----*  
//*  
//PDSR15 EXEC PGM=PDSR15  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//PACDBB DD DSN=&OLDBB,DISP=OLD  
//PACDB1 DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),DISP=(,CATLG,DELETE),  
// UNIT=&UNITS,VOL=&VOLS,  
// SPACE=&SPABB,  
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB  
//PACDIK DD SYSOUT=&OUT  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//SYSPRINT DD SYSOUT=&OUT  
//SYSUDUMP DD SYSOUT=&OUT  
//*
```


22.4. 'DR5J' PROCEDURE - DESCRIPTION OF STEPS

RETRIEVAL OF JOURNAL FILE: PDSR5J

.Permanent input file:
-Sequential image of journal file, Rel. 1.2 or 1.5
PACDBJ : DSN=&OLDBJ

.Permanent output file:
-Journal retrieved in the 2.5 format
DSN=&INDEXQ..&ROOT.&FILE.BJ(+1)

.Output report:
-Printing report
PACDIK

22.5. 'DR5J' PROCEDURE - EXECUTION JCL

```
//*****  
//* DSMS 2.5 *  
//*          RETRIEVAL OF SEQUENTIAL JOURNAL 1.2 OR 1.5      P *  
//*****  
//$RADP.DR5J PROC FILE=$FILE,          NUMBER OF PHYSICAL DATABASE  
//          ROOT=$ROOT,                ROOT OF DSMS SYSTEM  
//          OUT='$OUT',                 OUTPUT CLASS  
//          INDEQ='$INDEQ',            DATA GROUP FILE INDEX  
//          SPABJ=(TRK,(45,5),RLSE)',  SPACE OF BACKUP  
//          OLDBJ=,                    JOURNAL 1.2 OR 1.5  
//          VOLS='SER=$VOLO',          VOLUME OF GENERATION FILES  
//          UNITS='$UNITO',            UNIT OF GENERATION FILES  
//          STEPLIB='$MODB'            LIBRARY OF BATCH LM  
//*-----*  
//*  
//PDSR5J EXEC PGM=PDSR5J  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//PACDBJ DD DSN=&OLDBJ,DISP=OLD  
//PACDJB DD DSN=&INDEQ..&ROOT.&FILE.BJ(+1),DISP=(,CATLG,DELETE),  
//          UNIT=&UNITS,VOL=&VOLS,  
//          SPACE=&SPABJ,  
//          DCB=&INDEQ..DSCB.&ROOT.&FILE.BJ  
//PACDIK DD SYSOUT=&OUT  
//SYSOUT DD SYSOUT=&OUT  
//SYSOUX DD SYSOUT=&OUT  
//SYSPRINT DD SYSOUT=&OUT  
//SYSUDUMP DD SYSOUT=&OUT  
//*
```

VisualAge Pacbase - Operations Manual
DSMS - INSTALLATION & OPERATIONS
REPLACEMENT OF LOW-VALUES BY BLANKS

(DLVB)

PAGE 235

23

23. REPLACEMENT OF LOW-VALUES BY BLANKS (DLVB)

23.1. DLVB: REPLACEMENT OF LOW-VALUES BY BLANKS

REPLACEMENT OF LOW-VALUES BY BLANKS IN A BB FILE

The DLVB procedure inserts a blank wherever a low-value is present in the BB Database backup file.

The purpose of this procedure is to make possible the transfer of the BB file onto various platforms, while avoiding problems due to the presence of low-values during these transfers.

Utilization option

The DLVB procedure gives the user the opportunity to produce a transfer file containing only the 'data'-type records (refer to next subchapter).

In this case, the backup file obtained on the target platform after transfer will have to be reorganized (DREO procedure) in order to rebuild the cross-references (DX file).

EXECUTION CONDITIONS

None

23.2. DLVB: PARAMETERS-DESCRIPTION OF STEPS

DLVB: DESCRIPTION OF STEPS

REPLACEMENT OF LOW-VALUES BY BLANKS: PDSLVB

.EXEC line: Specify PARM='DATA' to keep only 'data'-type records in the output file.
To keep both 'index' and 'data' records, do not specify anything.

.Input file:

-Database backup

PACDBB : DSN=&INDEXQ..&ROOT.&FILE.BB(0)

.Output file:

-New Database backup

PACDB1 : DSN=&INDEXQ..&ROOT.&FILE.BB(+1)

23.3. DLVB: EXECUTION JCL

```
//*****  
//* DSMS 2.5 *  
//* REPLACEMENT OF LOW-VALUES BY BLANKS *  
//*****  
//$RADP.DLVB PROC ROOT='$ROOT', ROOT OF DSMS SYSTEM  
// FILE='$FILE', NUMBER OF THE DATABASE  
// INDEXQ='$INDEXQ', GENERATION-FILE INDEX  
// STEPLIB='$MOB', LIBRARY OF BATCH LM  
// OUT='$OUT', OUTPUT CLASS  
// UNITS='$UNITO', BACKUP UNIT (DISK OR TAPE)  
// SPABB='(TRK,(10,2),RLSE)' SPACE OF BACKUP (IF DISK)  
//*-----  
//PDSLVB EXEC PGM=PDSLVB,PARM=' '  
//STEPLIB DD DSN=&STEPLIB,DISP=SHR  
//SYSOUT DD SYSOUT=&OUT  
//PACDBB DD DSN=&INDEXQ..&ROOT.&FILE.BB(0),DISP=SHR  
//PACDB1 DD DSN=&INDEXQ..&ROOT.&FILE.BB(+1),  
// DISP=(,CATLG,DELETE),  
// UNIT=&UNITS,SPACE=&SPABB,  
// DCB=&INDEXQ..DSCB.&ROOT.&FILE.BB  
//SYSUDUMP DD SYSOUT=&OUT  
//*
```