VisualAge Pacbase 2.5

DSMS 2.5 IBM-CICS/OS/MVS
OPERATIONS MANUAL

DEDCI000251A

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.

First Edition (July 1998)

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.

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

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 procedures to be performed for standard reinstallation of corrected versions.

SITES WITH FORMER RELEASES

The new DSMS release, entirely compiled in Cobol II, requires a complete installation of the technical package, i.e. files, programs and batch procedures.

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

- Retrieval of DSMS 8.0.1 and adaptation to DSMS 2.5
- Retrieval of DSMS 8.02v01 or v02 and adaptation to DSMS 2.5
- Retrieval of DSMS 8.0.2 01 or 02 compatible with PACBASE 8.0.1 and adaptation to DSMS 2.5
- Retrieval of DSMS 1.2 or 1.5 as DSMS 2.5

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

2. DSMS COMPONENTS

2.1. INTRODUCTION

INTRODUCTION

DSMS requires the management of permanent data whether in batch or on-line mode (see the DSMS Reference Manual).

Three types of resources are necessary:

- Libraries which store the DSMS operating programs and the DSMS system parameters:

  . An on-line program library
  . A batch program library
  . A system parameter library

- Permanent files containing data manipulated by the DSMS system programs:

  . A system file containing error messages and HELP documentation,
  . User files containing the user and administrator data managed by DSMS.

- 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.

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

2.2. ON-LINE PROGRAM LIBRARY

ON-LINE PROGRAMS

PROGRAM CODE	ASSOCIATED CHOICE - COMMENTS
xxCHOI	Choice-decoding sub-program
xxTPDF	DAF extraction sub-program
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	Help function
xx00JO	JO
xx00KA	HK
xx00K1	LGKLAK
xx00K2	LPK
xx00K3	WS WU
xx00LE	LDELNCLSELDC LNC
xx00LS	LIE ....*...
xx00MA	H (Main Menu)
xx00PA	HP
xx00P1	PL
xx00QA	HQ
xx00QB	Q C ...... DD
xx00QC	R . ... CD
xx00Q1	Q . ...
xx00Q2	Q . ... D
xx00Q3	LCQ
xx00Q4	LVQ
xx00Q5	LJQ
xx00Q6	R . ...
xx00Q7	R L
xx00Q8	R C
xx00Q9	LCR

PROGRAM CODE	ASSOCIATED CHOICE - COMMENTS
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	Word-processor Upload/Download

PROGRAM CODE	Comments
xxCUAM	Sub-pgm for additional check on xx00B1
xxCUEV	Sub-pgm for additional check on xx00E1
xxCUMQ	Sub-pgm for additional check on xx00Q6
xxCURQ	Sub-pgm for additional check on xx00Q1
xxCUSI	Sub-pgm for additional check on xx00SI
xxMONI	Function's 'monitor' program
xxMOSO	Exit from DSMS transaction
xxUCTR	Management of the terminal's UCTRAN option for MVS/CICS ESA 3.1 releases
xxUCTX	Inhibition of the UCTRAN-option management from MVS/CICS ESA 3.2 releases onwards
xxUCTZ	Management of the terminal's UCTRAN option for use with MVS/CICS ESA 3.3. rel. onwards
xxSECT	Security Systems Interface

NOTE:

'xx' is the program prefix and corresponds to the ROOTD parameter (first 2 parameters of the chosen transaction code).

The size of the library is roughly 600 blocks, plus 30 'directory' blocks.

2.3. BATCH PROGRAM LIBRARY

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
PDSDAC	JCLDAF	Access sub-program for DAF
PDSBDF	-	Extraction sub-program for DAF
PDSEX	DEXT	DEXT flow monitor (french)
PDSEXE	DEXT	DEXT flow monitor (english)
PDSE90	DPRT	DPRT print sub-program
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	-	-
PDSUSI	-	-
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	-	
PDS900	DUPD	Update on DSMS database from DAF tables
REP2PJ	DEXQ	Extracts VA Pac journal < 2.0
PDS700	DXBJ	Extracts journalized transactions
PTU001	All	Copies entries onto a disk file
PDSR8B	DR80	Retrieval of an 8.0 / 8.01 DSMS database
PDSR8C		IMPORTANT: PACBASE MUST HAVE PREVIOUSLY
		BEEN UPDATED TO VERSION 8.0.2.
PDSR8X	DR8X	Retrieval of an 8.02 DSMS database
		compatible with PACBASE 8.01.
		TO BE USED SIMULTANEOUSLY WHEN SWITCHING
		OVER FROM PACBASE 8.01 TO PACBASE 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

MM1JCL		Parameterization of install./oper. JCL

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		Batch TOPSECRET
PACTSSC		CICS TOPSECRET

To use this extension, refer to chapter INSTALLATION, subchapter SECURITY SYSTEMS INTERFACE COMPLEMENT, and to the SECURITY SYSTEMS INTERFACE Reference Manual.

2.4. OPERATION-PARAMETER LIBRARY

THE SYSTEM PARAMETERS LIBRARY: SY

Its required size is approximately 5 blocks of 6,080 bytes.

It is a PDS file which contains the entries of the utilities used in the batch installation and operation procedures.

It includes the following:

.The DELETE/DEFINE's for the VSAM files :

MEMBERS	CONTENTS : DELETE/DEFINE OF THE FILE
DFxxy0DA	DSMS data (DA)
DFxxy0DC	VA Pac elements (DC)
DFxxy0DJ	DSMS journal (DJ)
DFxxy0DX	Cross-references (DX)
DFxx00DE	Error messages and HELP documentation (DE)
DFxxy0DF	On-line DAF work file

```
  xx = first two characters of the transaction code
  y  = third character of the transaction code
```

The physical characteristics of the files are described in the subchapters which follow.

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 manager.

.The VERIFY's and LISTCAT's of the VSAM files:

- VERIFff (ff = DA, DC, DJ, DX, DE) members contain the VERIFY PACDff command for each DSMS file.

- The LIxxy0DJ member contains the LISTCAT command for the Journal file (DJ).

. IDCAMS utility's input:

- MAXKEY: Maximum record

- REPRO999: Copy request

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

2.5. '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 : VSAM-KSDS
  .Recsize      : 90
  .Key          : 17 (position 0)
  .DSN          : &INDSV..&ROOTD.00DE
```

2.6. 'USER' FILES

USER FILES

These files contain user information and are managed by the system in either on-line or batch mode.

The first four contain the data directly managed by this function. They are:

.The DSMS Data file (DA)

```
     .Organization : VSAM-KSDS
     .Recsize      : 80 minimum, 350 maximum
     .CI size      : 4,096
     .Key          : 40 (position 2)
     .DSN          : &INDUV..&ROOTD.&ROOT2.0DA
```

.The Cross-References file (DX)

```
     .Organization : VSAM-KSDS
     .Recsize      : 80
     .CI size      : 4,096
     .Key          : 50 (position 0)
     .DSN          : &INDUV..&ROOTD.&ROOT2.0DX
```

.The VA Pac Elements file (DC)

```
     .Organization : VSAM-KSDS
     .Recsize      : 50 minimum, 168 maximum
     .CI size      : 4,096
     .Key          : 31 (position 2)
     .DSN          : &INDUV..&ROOTD.&ROOT2.0DC
```

.The DSMS Journal file (DJ)

```
     .Organization : VSAM-RRDS
     .Recsize      : 180
     .CI size      : 4,096
     .DSN          : &INDUV..&ROOTD.&ROOT2.0DJ
```

.The On-Line DAF work file (DF)

```
.Organization : VSAM-KSDS
.Recsize      : 100 min., 554 max.
.CI size      : 4,096
.Key          : 37 (position 2)
.DSN          : &INDUV..&ROOTD.&ROOT2.0DF
```

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          : &INDUN..&ROOTD.&ROOT2.0BB(n)
```

.The Archived Journal file (BJ)

```
.Organization : Sequential generation file
.Recsize      : 180
.DSN          : &INDUN..&ROOTD.&ROOT2.OBJ(n)
```

.The Deactivated Archived file (BQ)

```
.Organization : Sequential
.Recsize      : 180
.DSN          : User-defined
```

2.7. LIBRARY OF ENTRY-POINT SOURCES

THE LIBRARY OF ENTRY POINT SOURCES

```
. Size        : About 100 blocks of 6,080
. Organization: PDS
. DCB         : Recfm=FB;Lrecl=80;Blksize=6,080
. DSNAME      : &INDSN..&ROOTD.&ROOTD.SRC
```

This complementary library contains the user check sub-programs for the definitions of changes, events, sites, requests and layouts.

Member	Contents
xxCUAM	TP check on change definition
xxCUEV	TP check on event definition
xxCUMQ	TP check on layout definition
xxCURQ	TP check on request definition
xxCUSI	TP check on site definition
PDSCAM	Batch check on change definition
PDSCEV	Batch check on event definition
PDSCMQ	Batch check on layout definition
PDSCRQ	Batch check on request definition
PDSCSI	Batch check on site definition

It also contains the DAF tables' Dictionary (DAFDIC).

3. ENVIRONMENT

3.1. ON-LINE ENVIRONMENT

ON-LINE ENVIRONMENT

The monitor in use is CICS ESA, Release 3.1 or higher.

It must include BMS (the PAGING option is not used by DSMS).

The SPOOL=YES option is necessary to run batch jobs (choice LVQ).

All User files are updated on-line and must be protected by the 'DYNAMIC TRANSACTION BACKOUT' (ROLLBACK) option.

The use of the CICS 'EMERGENCY RESTART' option is recommended.

The maximum number of strings to be used per file is two. The value specified in the FCT can thus be adjusted without restriction according to the space available in the CICS partition.

GENERAL INFORMATION - HOW THE SYSTEM RUNS

The general characteristics are:

A single transaction code is used to manage the DSMS data. It points to a 'monitor' program whose purpose is to chain together the different programs that make up the system. No screen is directly driven by this program.

In case of a system ABEND, an 'ABEND MAP' is generated.

Updates are serialized; in other words, the system protects the concurrent accesses to the database by placing all update transactions in a wait queue (ENQUEUE and DEQUEUE in the update cycle).

EXIT FROM THE DSMS TRANSACTION

A special program is automatically called when you exit the DSMS transaction (program code: xxMOSO). This program performs an EXEC CICS RETURN, after displaying the "CURRENT DSMS CONVERSATION IS SAVED" if the exit is done by the '.12' option or by the corresponding function key (default: PF12).

If you wish the DSMS module not to branch off to CICS, you can replace this program with a user program containing for instance a call of user transaction.

DSMS sends a three-character COMMAREA to the program:

```
   Positions 1-2: Blank or 12 (for a save of the
                  conversation)
   Position 3:   'F' or 'A' according to the installation
                  language.
```

If you replace XXMODO by a program of your own, you can use the COMMAREA, and, if you find a '12' in positions 1-2, display the "CURRENT DSMS CONVERSATION IS SAVED" message (if an 'A' is in position 3), or the "LA CONVERSATION DSMS A ETE SAUVEGARDEE" message (if an 'F' is in position 3).

UPPERCASE AND LOWERCASE PROCESSING

The DSMS function has its own character management system:

. All codes entered in lowercase are automatically turned into uppercase,

. All clear names, labels and texts remain in lowercase.

FOR RELEASES SUBSEQUENT TO CICS ESA 3.2, THE UCTRAN OPTION CAN BE MANAGED AT THE TRANSACTION LEVEL.
THE DSMS TRANSACTIONS MUST HAVE THE UCTRAN=NO OPTION.
(In order to work only in uppercases, use the option UCTRAN=YES).

Under CICS ESA 3.1, the UCTRAN option of the terminal, if it is active, is AUTOMATICALLY INHIBITED by DSMS during all the connection. This allows to benefit from this management. (It is nevertheless possible to work in uppercases, by inhibiting the provided standard routine. To do so, rename the xxUCTR program then rename the xxUCTX Program into xxUCTR.

In CICS ESA Release 3.2 or higher, the UCTRAN option can be controlled at transaction level.

VA Pac transactions must have the UCTRAN=NO option. (to use uppercase letters only, set the UCTRAN option to YES).

In CICS ESA 3.1, and from CICS ESA 3.3, if the UCTRAN option of the terminal is active, it is automatically inhibited by VA Pac for the time of the connection, so that this control is available.

The installation process suggests 3 routine versions:

- The xxUCTR routine is to be used in a CICS ESA 3.1 context,

- the xxUCTZ routine (renamed xxUCTR) is to be used with a CICS ESA release higher or equal to 3.3,

- and the xxUCTX routine (renamed xxUCTR) is to be used with a CICS ESA release 3.2.

The latter can be used to deactivate the modification of the UCTRAN parameter in all CICS releases, which is useful if you wish to work always in uppercase letters.

3.2. BATCH ENVIRONMENT

THE BATCH ENVIRONMENT

In batch mode, the system runs using both the standard functions of the operating system and the VSAM access method.

The amount of memory needed for the execution of batch procedures varies according to the size of the buffers allocated to the files that use them.

3.3. ACCESS METHOD

ACCESS METHOD

The DSMS function manages its files using the indexed VSAM- KSDS access method without a secondary index, and the rela- tive VSAM-RRDS access method.

All files are protected against concurrent write accesses (SHARE OPTION 2).

All batch procedures include DELETE/DEFINE steps to take care of file reloading. Therefore, the files do not need to include the REUSE option and, consequently, can be allocated using either the UNIQUE option or the SUB-ALLOCATION option.

4. BATCH PROCEDURES

4.1. INTRODUCTION

INTRODUCTION

Batch processing with DSMS is divided into various procedures which are described in the following chapters with their specific execution conditions.

For each procedure, there is:

. A general introduction containing:

- The presentation,
- The execution condition(s),
- The action to be taken in case of abnormal execution.

. The description of user input, processes, results, and possible recommendations.

. A presentation of each step containing:

- The files used (temporary and permanent),
- The return codes generated by the procedure (where applicable).

4.2. CLASSIFICATION OF PROCEDURES

PROCEDURE CLASSIFICATION

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-reference 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 batch procedure DUPT.

. Printing of query results, and printing of table and keyword lists (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 developer's workstation (revamped version) (DEXH).

. Batch update of DSMS files of events, changes, sites or tables (DUPT, DUPD).

. Pre-processing of DAF source failes (DPDF).

. Renaming of table, site and keyword codes (DREN).

. Printing of the list of installed programs (DINS).

RETRIEVAL OF PREVIOUS RELEASES

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

PREVIOUS-RELEASE RETRIEVAL PROCEDURES

. Retrieval of DSMS 8.0.1 database (DR80).

. Retrieval of DSMS 8.0.2 database COMPATIBLE with PACBASE 8.0.1 (DR8X) to be used when switching over from PACBASE 8.0.1 to 8.0.2.

. Retrieval of DSMS 8.0.2 01 or 02 database (DR8Q) (retrieval of queries).

. Retrieval of DSMS 1.2 or 1.5 database (DR15)

. Retrieval of DSMS 1.5 journal archived (DR5J)

RETRIEVAL OF A DATABASE FOR ANOTHER PLATFORM

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

4.3. ABNORMAL EXECUTION

ABNORMAL EXECUTIONS

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

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

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

In most cases, examining the status and type of operation allows the user to find the cause of the abnormal end.

The reader will find in the charts below the most common values for the status and type of operation.

NN	STATUS
21	SEQUENCE ERROR
22	DUPLICATE KEY
23	NO RECORD FOUND
24	BOUNDARY VIOLATION (KSDS-RRDS)
30	SYSTEM ERROR
34	BOUNDARY VIOLATION(SEQUENTIAL)
92	LOGIC ERROR (FOR EXAMPLE, OPEN OF AN ALREADY OPEN FILE)
93	FILE STILL OPEN UNDER CICS
95	INVALID OR INCOMPLETE FILE INFORMATION

OO	OPERATION
W	WRITE
RW	REWRITE
RU	READ UP
OP	OPEN
CL	CLOSE
D	DELETE
R	READ
P	START
RN	READ NEXT

When this message is absent, and the type of ABEND generated directly signals a problem in the VisualAge Pacbase system programs, it will be necessary to contact the VisualAge Pacbase technical team at IBM. Be sure to KEEP ALL LISTINGS that may be necessary to analyze the problem.

5. JOURNAL ARCHIVING (DARC)

5.1. INTRODUCTION

DARC: INTRODUCTION

The Journal Archive procedure (DARC) backs-up the Journal file (DJ) as a sequential file (BJ), and re-initializes it both logically and physically.

The new archived transactions do not override 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 processing.

ABNORMAL EXECUTION

Refer to Subchapter 'ABNORMAL EXECUTIONS' in Chapter "THE BATCH PROCEDURES".

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

Otherwise, the procedure should be restarted after modification of user input in order to specify a re-initialization request without a backup of the Journal file (DJ), since it has already been backed up.

CAUTION:

With systems using generation files (MVS for instance), the +1 version of the archived transaction file could 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 optional input for:

. deactivating previously archived transactions that are now obsolete,

. indicating the absence of previously archived transac- tions during input,

. indicating the unavailability of the Data file (DA) during input,

. requesting only a re-initialization of the transaction file.

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 independent of each other and must be treated as such. They are ignored if it is indicated that there are no previously archived transactions.

The unavailability of the Data file is indicated only when this file has been physically destroyed (see paragraph "RECOMMENDATIONS").

The re-initialization request without an archive is necessary when the Journal file is physically destroyed.

NOTE: In this case, the previous archive is not duplicated on the output archive. When the cataloging is automatic, previous archives 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 archive 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 coherent state, and if the Journal file is correctly formatted.

When a database needs to be restored after any kind of problem, some information in the database may be destroyed and it is not possible to execute the DARC nor the DRST procedures.

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

. If the Data file (DA) is lost or has been flagged as 'incoherent', 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 incoherent.

. If the Journal file (DJ) is lost or destroyed, a 'J' must be entered in column 17. The DARC procedure will format an empty Journal file. Then, the DRST procedure may be executed.

. If the Archived Journal file (BJ) is lost or destroyed, a '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 archived transactions is obtained.

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

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 archival,
- 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=&INDUN..&ROOTD.&ROOT2.0BJ(0)
 -Journal file to re-initialize
  PACDDJ  :    DSN=&INDUV..&ROOTD.&ROOT2.0DJ
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Input-Output file:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA

.Output files:
 -Archived update transactions
  PACDBJ  :    DSN=&INDUN..&ROOTD.&ROOT2.0BJ(+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.
```

RE-INITIALIZATION OF THE JOURNAL FILE: PDS320

This step executes the following:

- Creates the first record in the Journal file

- Repositions the Data file flag.

```
.Input files:
 -User transaction
  PACDMB  :    DSN=&&PAC7MB
 -Error-message file (PACDDE)
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Input-Output file:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA

.Output file:
 -Journal file to be re-initialized
  PACDDJ  :    DSN=&INDUV..&ROOTD.&ROOT2.0DJ

.Output report:
 -Review of reinitialization
  PACDRU
```

5.4. EXECUTION JCL

```
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*                  - ARCHIVAL OF DSMS JOURNAL -                      *
//**********************************************************************
//$RADP.DARC PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SPACE
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  VOLS='SER=$VOLUN',     VOLUME OF ARCHIVED JOURNAL (BJ)
//  UNITS='$UNITUN',       ARCHIVED JOURNAL UNIT (DISK OR TAPE)
//  UWK=$UWK,              WORK UNIT
//  SPABJ='(TRK,(10,2),RLSE)'   SPACE OF JOURNAL (IF DISK)
//**********************************************************************
//* INPUT       : COMMAND FOR DEACTIVATION OF ARCHIVED TRANSACTIONS    *
//* COL  2      : 'S'                                                  *
//* COL  3 A  6 : SESSION NUMBER                                       *
//* COL  7 A 14 : DATE (CCYYMMDD)                                      *
//* COL  15     : ' ' PRESENCE OF INPUT ARCHIVED TRANSACTION FILE      *
//*             : 'I' ABSENCE OF INPUT ARCHIVED TRANSACTION FILE       *
//* COL  16     : ' ' PRESENCE OF DATA FILE (DA)                       *
//*             : 'D' ABSENCE OF DATA FILE (DA)                        *
//* COL  17     : ' ' ARCHIVAL AND REINITIALIZATION                    *
//*             : 'J' REINITIALIZATION WITHOUT ARCHIVAL                *
//*                                                                    *
//* IN THE ABSENCE OF INPUT (OR ERROR ON A COMMAND PARAMETER),         *
//* NO DEACTIVATION WILL TAKE PLACE, HOWEVER ARCHIVAL AND              *
//* REINITIALIZATION WILL BE EXECUTED NORMALLY.                        *
//*                                                                    *
//* TRANSACTIONS WHOSE SESSION (DATE) IS PRIOR OR EQUAL TO             *
//* THE SESSION (DATE) INDICATED ARE NOT KEPT. THEY ARE                *
//* RECOVERED IN THE FILE OF DEACTIVATED TRANSACTIONS.                 *
//*                                                                    *
//**********************************************************************
//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=3200)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDJ),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDS300 EXEC PGM=PDS300,REGION=2048K
//***********************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
```

```
//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)
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,PASS)
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDJB   DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(0),DISP=OLD
//PACDBJ   DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//        SPACE=&SPABJ,
//        DCB=(&DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//PACDBQ   DD DUMMY,DCB=BLKSIZE=180
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELDEF EXEC PGM=IDCAMS,COND=(0,NE,PDS300)
//*****************************************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..&ROOTD.&ROOTD.SY(DF&ROOTD.&ROOT2.0DJ),DISP=SHR
//PDS320 EXEC PGM=PDS320,COND=(0,NE,PDS300)
//*****************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,PASS)
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELBJ  EXEC PGM=IEFBR14,COND=(08,NE,PDS300)
//******************************************
//DDBJ     DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(+1),DISP=(OLD,DELETE)
```

6. PRINTING OF QUERY OUTPUT (DPRT)

6.1. INTRODUCTION

DPRT: INTRODUCTION

The DPRT procedure performs all the printing operations for DSMS:

. Results of DSMS Queries on Events, Changes and Sites, (this order must be respected)

. Printouts of Tables, Keywords, Queries and Layouts.

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 CONDITION

None, the Database can remain open to on-line processing.

ABNORMAL EXECUTION

Refer to Chapter "THE BATCH PROCEDURES", Subchapter 'ABNORMAL EXECUTIONS'.

6.2. INPUT - PROCESSING - RESULTS

USER INPUT

A compulsory '*' line:

Col.	Len.	Value	Description
2	1	'*'	Line Code
3	8	uuuuuuuu	DSMS User Code
11	8	pppppppp	Password
19	3	ppp	Product Code
22	2	su	Subsidiary Code
24	1	l	Language Code

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

Col.	Len.	Value	Description
TABLES			
02	03	Txx	Table codes for Txx (except TRA, TUD, TUG, TUP and TUS)
06	02	C1	... with their label in connected user language (default option)
		C2	... with all labels
02	03	TUD	User codes with all authorizations (TUG, TUP, and TUS)

Col.	Len.	Value	Description
KEYWORDS			
02	04	LAKC	Stand-alone Keywords for Changes
		LPKC	Principal keywords for Changes
		LGKC	All keywords for Changes
06	01	l	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
		LAKT	Stand-alone Techn. Keywords for Evnts
		LPKT	Principal Techn. Keywords for Events
		LGKT	All Technical Keywords for Events

Col.	Len.	Value	Description
QUERIES / REPORTS			
02	04	X QC	Query on Changes
		X QE	Query on Events
		X QS	Query on Sites
		X RC	Report on Changes
		X RE	Report on Events
		X RS	Report on Sites
06	06	xxxxxx	Query or Report code
12	08	uuuuuuuu	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
02	04	LCQC	Queries on Changes
		LCQE	Queries on Events
		LCQS	Queries on Sites
		LCRC	Reports on Changes
		LCRS	Reports on Sites
		LCRE	Reports on Events
		LCQ	All Queries
		LCR	All Reports
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	uuuuuuuu	User code for Queries/Reports owner

Col.	Len.	Value	Description
			.PRINT VIA USER QUERY:
5	6	rrrrrr	Code of the user query (compulsory) "Q" Entity used.
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.
			.PRINT OF QUERIES/LAYOUTS:
6	6	xxxxxx	Query/Layout code (Compulsory for X type of extractions)
12	8	uuuuuuuu	Code of the user who made the query/ Layout (optional), default value: user corresponding to the '*' line code.

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 Layouts.

RETURN CODE

0	OK with queries
4	OK with table, kyw., query, or report list requests
8	OK with erroneous querie or other request
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 flow monitor for the different programs, which are therefore sub-programs of this monitor.
The procedure includes the following steps:
The input file is automatically formatted when QUERIES are entered in on-line mode and the JOB submitted in the same mode.

```
PRINTING:  PDSB

.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -VA Pac-element file
  PACDDC  :    DSN=&INDUV..&ROOTD.&ROOT2.0DC
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Input file:
 -User queries
  PACDMB  :    DSN=&&PACDMB

.Work files:
 -Print requests
  PACDKD
 -Requests
  PACDKQ
 -Temporary files
  PACDID
 -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/layouts
  PACDRQ
 -Print of check cards
  PACDJQ
```

6.4. EXECUTION JCL

```
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*              - PRINTOUTS AND QUERIES -                             *
//**********************************************************************
//$RADP.DPRT PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORTLIB
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SIZE
//  SPAMB='(TRK,(30,5),RLSE)', SPACE OF EXTRACTION COMMANDS
//  COPIES=1,              NUMBER OF REPORT COPIES
//  SPAWK='(CYL,(20,2))',  WORK FILE SPACE
//  OUT='$OUT',            OUTPUT CLASS
//  OUTL='$OUT',           QUERY OUTPUT CLASS
//  LNG=E,                 LANGUAGE OF MONITOR (E FOR ENGLISH)
//  UWK=$UWK               WORK UNIT
//**********************************************************************
//* INPUT :                                                            *
//* .. IDENTIFICATION LINE                                             *
//*  COL 02     : *                                                    *
//*  COL 03     : DSMS USER CODE                                       *
//*  COL 11     : PASSWORD                                             *
//*  COL 19-21  : PRODUCT CODE                                         *
//*  COL 22-23  : SUBSIDIARY CODE                                      *
//*  COL 24     : LANGUAGE CODE                                        *
//*                                                                    *
//*  ..  EXTRACT COMMAND LINE(S)                                       *
//*      ----------------------                                        *
//*  COL 02-05  : TYPE OF EXTRACTION                                   *
//*  -- EXTRACTION BY USER QUERY :                                     *
//*  COL 05-10  : QUERY CODE                                           *
//*  COL 17     : DELIMITER                   <--- OPTIONAL            *
//*  COL 18     : SYMBOL                      <--- OPTIONAL            *
//*  COL 19     : SEPARATOR                   <--- OPTIONAL            *
//*  COL 20-73  : PARAMETERS VALUES           <--- OPTIONAL            *
//*  -- EXTRACTION OF QUERIES/LAYOUT :                                 *
//*  COL 06-11  : QUERY OR LAYOUT CODE                                 *
//*  COL 12-19  : OWNER OF THE QUERY/LAYOUT   <--- OPTIONAL            *
//**********************************************************************
//INPUT  EXEC PGM=PTU001
//***********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&PACDMB,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=1600,SPACE=&SPAMB
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//VERIFY EXEC PGM=IDCAMS
//***********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDSB     EXEC PGM=PDSB&LNG,REGION=4096K
//*************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
```

```
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,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
//PACDQR   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6650
//PACDQJ   DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=6650
//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
//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
```

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.

These modifications must be made in the System Parameters library.

EXECUTION CONDITION

The database must be closed to on-line processing.

Since this procedure re-creates the database, it is recommended to have previously readjusted the sizes of the different database files according to their estimated evolution.

These modifications must be made in the System Parameters library (SY).

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 EXECUTIONS' in Chapter "THE BATCH PROCEDURES".

Whatever the cause, the procedure can be restarted as it is after the problem is solved.

NOTE:

Since the DRST procedure recreates the files, it may be useful to adjust the files' sizes according to their estimated growth.

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" 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 modified.

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	'l'	Language code 'E' or 'F' (optional)
4	1		Journal inhibition flag
		'0'	No inhibition (default option)
		'1'	Inhibition
5	1		This field may only be used with DOS/VSE
		'I'	Default option for all hardware
		'N'	DOS/VSE: if CURRENT-DATE = DD/MM/CCYY
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 firt 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 in the same as a 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 layout 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

REPORT

This procedure prints a report listing the requested options, any associated errors, the number of records restored in the database for each file, and the options stored 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 archived transaction retrieval has been requested.

7.3. DESCRIPTION OF STEPS

DRST: DESCRIPTION OF STEPS

```
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
  PACDDJ  :    DSN=&INDUV..&ROOTD.$ROOT2.0DJ
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.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 are executed.



DATABASE RESTORATION:  PDS400

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

.Permanent input files:
 -Backup of the files
  PACDBB  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(0)
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Permanent output files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -VA Pac-element file
  PACDDC  :    DSN=&INDUV..&ROOTD.&ROOT2.0DC
 -Journal file
  PACDDJ  :    DSN=&INDUV..&ROOTD.&ROOT2.0DJ
 -Cross-reference file
  PACDDX  :    DSN=&INDUV..&ROOTD.&ROOT2.0DX
```

```
.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
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -VA Pac-element file
  PACDDC  :    DSN=&INDUV..&ROOTD.&ROOT2.0DC
 -Cross reference file
  PACDDX  :    DSN=&INDUV..&ROOTD.&ROOT2.0DX

.Input files:
 -Work file      (2 records)
  PACDMS  :    DSN=&&PACDMS
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Input archived file:
 -Arch. of the journal to retrieve
  PACDBJ  :    DSN=&INDUN..&ROOTD.&ROOT2.0BJ(0)

.Output report:
 -Update report
  PACDRU
```

7.4. EXECUTION JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - LOADING-RESTORATION OF DSMS DATABASE -             *
//*********************************************************************
//$RADP.DRST PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  UWK='$UWK'             WORK UNIT
//*********************************************************************
//*  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 FOR MM/DD/YY)               *
//*              :                     (N FOR DD/MM/YY)               *
//*   COL 06-08  : REC : RETRIEVAL OF ARCHIVED TRANSACTIONS           *
//*   COL 09-20  : (NOT USED)                                         *
//*   COL 21     : SECURITY SYSTEM (R,S, ,&)                          *
//*   COL 22     : USER CONTROL UNDER RACF (N, ,&)                    *
//*   COL 23     : CRYPT/UNCRYPT OF PASSWORD (C,D, )                  *
//*   COL 24-25  : (NOT USED)                                         *
//*   COL 26     : CALL OF SUB-PGM FOR CHANGES (C, ,&)                *
//*   COL 27     : CALL OF SUB-PGM FOR EVENTS (E, ,&)                 *
//*   COL 28     : CALL OF SUB-PGM FOR QUERIES (Q, ,&)                *
//*   COL 29     : CALL OF SUB-PGM FOR LAYOUTS (R, ,&)                *
//*   COL 30     : CALL OF SUB-PGM FOR SITES (S, ,&)                  *
//*                                                                   *
//*  IF THE JOURNAL FILE OF TRANSACTIONS ON DISK (DJ) IS NOT          *
//*  REINITIALIZED, NO RESTORATION IS 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=&&RESTMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//EXISDJ EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..&ROOTD.&ROOTD.SY(LI&ROOTD.&ROOT2.0DJ),DISP=SHR
//PDS380 EXEC PGM=PDS380,COND=(0,NE,EXISDJ)
//*****************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRU   DD SYSOUT=&OUT
//DEFINE EXEC PGM=IDCAMS,COND=(0,NE,PDS380)
//*****************************************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..&ROOTD.&ROOTD.SY(DF&ROOTD.&ROOT2.0DA),DISP=SHR
//      DD DSN=&INDSN..&ROOTD.&ROOTD.SY(DF&ROOTD.&ROOT2.0DC),DISP=SHR
//      DD DSN=&INDSN..&ROOTD.&ROOTD.SY(DF&ROOTD.&ROOT2.0DJ),DISP=SHR
//      DD DSN=&INDSN..&ROOTD.&ROOTD.SY(DF&ROOTD.&ROOT2.0DX),DISP=SHR
```

```
//PDS400 EXEC PGM=PDS400,COND=(0,NE,PDS380)
//****************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDMB   DD DSN=&&RESTMB,DISP=(OLD,PASS)
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMS   DD DSN=&&PACDMS,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PACDRU   DD SYSOUT=&OUT
//PACDBB   DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(0),DISP=OLD
//SYSUDUMP DD SYSOUT=&OUT
//PDS450 EXEC PGM=PDS450,COND=((0,NE,PDS380),(0,NE,PDS400))
//*********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDBJ   DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(0),DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMS   DD DSN=&&PACDMS,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

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 use in order to ensure its coherence during execution of the DSAV procedure.

ABNORMAL EXECUTION

Refer to Subchapter 'ABNORMAL EXECUTIONS' in Chapter "THE BATCH PROCEDURES".

The main cause of an abnormal end of the DSAV procedure 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.

GENERAL RESULTS

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

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

NOTE: The DSAV procedure increments the current session number.

8.3. DESCRIPTION OF STEPS

```
DATABASE CONSISTENCY CHECK:  PDSBAS

.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Output report
 -Validity report (Length=079)
  PACDRS

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

DSMS BACKUP:  PDS500

.Input-Output file:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA

.Permanent Input files:
 -VA Pac-element file
  PACDDC  :    DSN=&INDUV..&ROOTD.&ROOT2.0DC
 -Cross-reference file
  PACDDX  :    DSN=&INDUV..&ROOTD.&ROOT2.0DX
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

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

.Output file:
 -Sequential image of files
  PACDBB  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1)

.Output report:
 -Backup report
  PACDRU
```

8.4. EXECUTION JCL

```
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*              - BACKUP OF THE DSMS DATABASE -                       *
//**********************************************************************
//$RADP.DSAV PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  UWK='$UWK',            WORK UNIT
//  SPABB='(TRK,(10,2),RLSE)'   SPACE OF BACKUP (IF DISK)
//**********************************************************************
//* INPUT                                                              *
//*  COL 02     : 'O'                                                  *
//*  COL 03-05  : CRYPT, UNCRYPT (ENC,DEC,   )                         *
//*                                                                    *
//**********************************************************************
//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=6160)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDX),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDSBAS EXEC PGM=PDSBAS
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDS500 EXEC PGM=PDS500,COND=(0,NE,PDSBAS)
//*****************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMB   DD DSN=&&DSAVMB,DISP=(OLD,PASS)
//PACDBB   DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPABB,
//            DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELBB  EXEC PGM=IEFBR14,COND=(08,NE,PDS500)
//*****************************************
//DDBB     DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),DISP=(OLD,DELETE)
```

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 will be 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 CONDITION

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

The updates executed after the backup file used for reorganization is rebuilt, can be retrieved during the restoration of the reorganized database.

ABNORMAL EXECUTION

Refer to Subchapter 'ABNORMAL EXECUTIONS' 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.

The format of this input is provided 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 code	(20 x 3 char.) If Col.2 = 'P'
	60	Subsid. code	(30 x 2 char.) If Col.2 = 'S'
	60	Prod./ Subsid.	(12 x 5 char.) If Col.2 = 'X'

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 which may be used as input to the Restoration (DRST) procedure.

9.3. DESCRIPTION OF STEPS

```
BUILDING OF INDEXES (except keywords):  PDSR10

.Input file:
 -Input file
  CARTE  :    DSN=&&PACDMB

.Permanent Input files:
 -DSMS database backup
  PACDBB  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(0)
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Work files:
 -Data and VA Pac elements
  PACDX1  :    DSN=&&W1
 -Keywords and keyword references
  PACDW2  :    DSN=&&W2
 -Cross-references (except keywords)
  PACDW3  :    DSN=&&W3
 -Sort files
  SORTWK01, 02, 03

.Output reports:
 -Inconsistencies among 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
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Permanent output file:
 -Reorganized DSMS database backup
  PACDBB  :    _SN=&INDUN..&ROOTD.&ROOT2.0BB(+1)

.Output report:
 -Reorganization report
  PACDRR
```

9.4. EXECUTION JCL

```
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*              - DSMS REORGANIZATION -                               *
//**********************************************************************
//$RADP.DREO PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  CYL='(4,1)',           SORTWORK SPACE
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  UWK=$UWK,              WORK UNIT
//  SPADA='(TRK,(60,5),RLSE)',  WORK SPACE (DA + DC)
//  SPADX='(TRK,(60,5),RLSE)',  WORK SPACE (DX)
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  SPABB='(TRK,(10,2),RLSE)'   SPACE OF BACKUP (IF DISK)
//**********************************************************************
//* OPTIONAL INPUT                                                     *
//*  COL 02     : DELETION OF PRODUCTS, SUBSIDIARIES OR                *
//*               PRODUCT/SUBSIDIARY ENVIRONMENT (P,S,X)               *
//*  COL 03-62  : 20 PRODUCTS, 30 SUBSIDIARIES OR                      *
//*               12 PRODUCT/SUBSIDIARY ENVT                           *
//*                                                                    *
//**********************************************************************
//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=3200)
//PDSR10  EXEC PGM=PDSR10
//***********************
//STEPLIB   DD DSN=&STEPLIB,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//PACDBB    DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(0),DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRH    DD SYSOUT=&OUT
//PACDRK    DD SYSOUT=&OUT
//CARTE     DD DSN=&&PACDMB,DISP=(OLD,DELETE)
//PACDW1    DD DSN=&&W1,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=VB,BLKSIZE=6022,LRECL=354)
//PACDW2    DD DSN=&&W2,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADX,
//          DCB=(RECFM=FB,BLKSIZE=6240,LRECL=120)
//PACDW3    DD DSN=&&W3,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=FB,BLKSIZE=6400,LRECL=80)
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//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
//PDSR20  EXEC PGM=PDSR20
//***********************
//STEPLIB   DD DSN=&STEPLIB,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//PACDW2    DD DSN=&&W2,DISP=(OLD,PASS)
//PACDW4    DD DSN=&&W4,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADX,
//          DCB=(RECFM=FB,BLKSIZE=6120,LRECL=340)
//PACDW5    DD DSN=&&W5,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=FB,BLKSIZE=6400,LRECL=80)
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//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
//PACDW3    DD DSN=&&W3,DISP=(OLD,PASS)
//PACDW5    DD DSN=&&W5,DISP=(OLD,PASS)
//PACDW6    DD DSN=&&W6,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPADA,
//          DCB=(RECFM=FB,BLKSIZE=6400,LRECL=80)
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//SYSUDUMP  DD SYSOUT=&OUT
//PDSR40  EXEC PGM=PDSR40
//***********************
//STEPLIB   DD DSN=&STEPLIB,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDBB    DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDRR    DD SYSOUT=&OUT
//PACDW1    DD DSN=&&W1,DISP=(OLD,PASS)
//PACDW4    DD DSN=&&W4,DISP=(OLD,PASS)
//PACDW6    DD DSN=&&W6,DISP=(OLD,PASS)
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUX    DD SYSOUT=&OUT
//SYSOUT    DD SYSOUT=&OUT
//SYSUDUMP  DD SYSOUT=&OUT
//DEL12  EXEC PGM=IEFBR14,COND=(12,NE,PDSR40)
//******************************************
//DDBB     DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),DISP=(OLD,DELETE)
```

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 into transactions to be used as input to the DSMS Database Updating (DUPD) procedure, in order to update the corresponding modified elements.

EXECUTION CONDITION

None.

ABNORMAL EXECUTION

Refer to Subchapter 'ABNORMAL EXECUTIONS' in Chapter "THE BATCH PROCEDURES".

If an abnormal end occurs, the procedure can be restarted with no additional modifications 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 transactions
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

```
TRANSACTION EXTRACTION AND FORMATTING:  PDS600

.Permanent imput files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE
 -VA Pac archived transactions
  PAC7PJ  :    DSN=&PAC7PJ

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

.Sort files:
 -SORTWK01, 02, 03

.Output file:
 -Update transaction file for DUPT
  PACDMV  :    DSN=&&PACDMV

.Output report:
 -Report on selection request
  PACDRU

.Return codes:
 - 0:  No error
 -04:  No error and printout requested
 -08:  Error on the user line code or on a parameter
 -12:  I/O error on a file


PRINTING OF DSMS UPDATE TRANSACTIONS:  PDS610

.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Input File:
 -Update transaction 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

```
//*******************************************************************
//* VisualAge Pacbase-DSMS                                          *
//*                                                                 *
//*              - EXTRACTION-UPDATE OF DSMS DATABASE -             *
//*******************************************************************
//$RADP.DEXP PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SPACE
//  SPAMV='(TRK,(1,1))',   SPACE OF EXTRACTED TRANSACTIONS
//  PAC7PJ='NULLFILE',     DSN OF ARCHIVED TRANSACTIONS FILE
//  OUT='$OUT',            OUTPUT CLASS
//  UWK='$UWK'             WORK UNIT
//*******************************************************************
//*   FORMAT OF TRANSACTIONS AT INPUT :                             *
//*   .. A DSMS USER AND PASSWORD LINE                              *
//*  COL 02     : *                                                 *
//*  COL 03     : DSMS USER CODE                                    *
//*  COL 11     : PASSWORD                                          *
//*   .. 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 PACBASE DATABASE CODE                    *
//*  COL 36-38  : LIBRARY CODE                                      *
//*  COL 39-54  : TYPE OF ENTITIES TO BE SELECTED                   *
//*  COL 55     : EXTRACT OF TRANSAC. FOR CHANGE 999999 ( ,N)       *
//*  COL 56     : PRINTING OF ALL TRANSACTIONS          ( ,N)       *
//*  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=&&PACDMB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDS600 EXEC PGM=PDS600
//*********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
```

```
//SYSOUX   DD SYSOUT=&OUT
//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)
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,PASS)
//PACDMV   DD DSN=&&PACDMV,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAMV,
//         DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PAC7PJ   DD DSN=&PAC7PJ,DISP=SHR
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDS610 EXEC PGM=PDS610,COND=(4,NE,PDS600)
//*****************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMV   DD DSN=&&PACDMV,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

10.5. PROCEDURE JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - EXTRACTION-UPDATE OF DSMS DATABASE -               *
//*********************************************************************
//$RADP.DEXQ PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SPACE
//  SPAMV='(TRK,(1,1))',   SPACE OF EXTRACTED TRANSACTIONS
//  PAC7PJ='NULLFILE',     DSN OF ARCHIVED TRANSACTIONS FILE
//  SPAJP='(TRK,(1,1))',   SPACE OF NEW PACBASE JOURNAL
//  OUT='$OUT',            OUTPUT CLASS
//  UWK='$UWK'             WORK UNIT
//*********************************************************************
//*   FORMAT OF TRANSACTIONS AT INPUT :                               *
//*   .. A DSMS USER AND PASSWORD LINE                                *
//*  COL 02     : *                                                   *
//*  COL 03     : DSMS USER CODE                                      *
//*  COL 11     : PASSWORD                                            *
//*   .. 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 PACBASE DATABASE CODE                      *
//*  COL 36-38  : LIBRARY CODE                                        *
//*  COL 39-54  : TYPE OF ENTITIES TO BE SELECTED                     *
//*  COL 55     : EXTRACT OF TRANSAC. FOR CHANGE 999999 ( ,N)         *
//*  COL 56     : PRINTING OF ALL TRANSACTIONS          ( ,N)         *
//*  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=&&PACDMB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//REP2PJ EXEC PGM=REP2PJ
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
```

```
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PAC7PJ   DD DSN=&PAC7PJ,DISP=SHR
//PAC7JP   DD DSN=&&PAC7JP,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAJP,
//         DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
//PDS600 EXEC PGM=PDS600
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//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)
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,PASS)
//PACDMV   DD DSN=&&PACDMV,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAMV,
//         DCB=(RECFM=FB,LRECL=250,BLKSIZE=6250)
//PAC7PJ   DD DSN=&&PAC7JP,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDS610 EXEC PGM=PDS610,COND=(4,NE,PDS600)
//*****************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMV   DD DSN=&&PACDMV,DISP=(OLD,PASS)
//PACDRU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

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 Updating (DUPT) procedure.

PRINCIPLE

In order to select the extraction of Changes, Events or Sites, the procedure uses Queries ("Q" entities) that must be 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 map ("R" entity) associated with the Query used for the extraction does not interfere in the extraction.

EXECUTION CONDITION

None.

ABNORMAL EXECUTION

Refer to Subchapter 'ABNORMAL EXECUTIONS' in Chapter: THE BATCH PROCEDURES.

If an abnormal end occurs, the procedure can be restarted with no additional modifications 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	uuuuuuuu	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 / LAYOUTS:			
02	04	X QC	Query on Changes
		X QE	Query on Events
		X QS	Query on Sites
02	04	X RC	Layout on Changes
		X RE	Layout on Events
		X RS	Layout on Sites
12	08	uuuuuuuu	Owner of the Query or Layout (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	Layouts on Changes
		LCRE	Layouts on Events
		LCRS	Layouts on Sites
12	08	uuuuuuuu	Owner of Queries or Layouts
KEYWORDS:			
02	04	LAKC	Isolated keywords of Changes
		LGKC	All Changes' Keywords
06	01	l	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 BY USER QUERY:
5	6	rrrrrr	User Query code (required)
			- Used 'Q' Entities
5	6	mmmmmm	Layout code (optional)
17	1	d	Delimiter (optional)
			Parameter-settings:

18	1	s	Symbol -
19	1	x	Separator -
20	54	.....	Parameter values -
			If optional some optional fields were not completed, default values will be used. Theyr 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 parameter
12	I/O error on a file or Inconsistencies among DSMS data
16	Sort error

11.3. DESCRIPTION OF STEPS

DEXT: DESCRIPTION OF STEPS

This procedure calls a unique 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:

```
EXTRACTIONS:  PDSEX

.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -VA Pac element file
  PACDDC  :    DSN=&INDUV..&ROOTD.&ROOT2.0DC
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.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

```
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*              - EXTRACTION OF BATCH TRANSACTIONS FOR DUPT -         *
//**********************************************************************
//$RADP.DEXT PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORTLIB
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SIZE
//  SPAIM='(TRK,(150,15),RLSE)', SPACE OF EXTRACTED TRANSACTIONS
//  SPAMB='(TRK,(150,15))', SPACE OF EXTRACTION COMMANDS
//  SPAWK='(CYL,(20,2))',  WORK FILE SPACE
//  OUT='$OUT',            UTILITIES OUTPUT CLASS
//  OUTL='$OUT',           OUTPUT CLASS OF REPORTS
//  LNG=E,                 LANGUAGE OF MONITOR (E FOR ENGLISH)
//  UWK=$UWK               WORK UNIT
//**********************************************************************
//* INPUT :                                                            *
//* .. IDENTIFICATION LINE                                             *
//*  COL 02     : *                                                    *
//*  COL 03     : DSMS USER CODE                                       *
//*  COL 11     : PASSWORD                                             *
//*  COL 19-21  : PRODUCT CODE                                         *
//*  COL 22-23  : SUBSIDIARY CODE                                      *
//*  COL 24     : LANGUAGE CODE                                        *
//*                                                                    *
//*  ..  EXTRACT COMMAND LINE(S)                                       *
//*      -----------------------                                       *
//*  COL 02-05  : TYPE OF EXTRACTION                                   *
//*  -- EXTRACTION BY USER QUERY :                                     *
//*  COL 05-10  : QUERY CODE                                           *
//*  COL 17     : DELIMITER                   <--- OPTIONAL            *
//*  COL 18     : SYMBOL                      <--- OPTIONAL            *
//*  COL 19     : SEPARATOR                   <--- OPTIONAL            *
//*  COL 20-73  : PARAMETERS VALUES           <--- OPTIONAL            *
//*  -- EXTRACTION OF QUERIES/LAYOUT :                                 *
//*  COL 06-11  : QUERY OR LAYOUT CODE                                 *
//*  COL 12-19  : OWNER OF THE QUERY/LAYOUT   <--- OPTIONAL            *
//*                                                                    *
//**********************************************************************
//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=3200)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDSEX    EXEC PGM=PDSEX&LNG
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
```

```
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,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
//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
```

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 will be able to create 'external value lists' that will be used by the revamped DSMS workstation.

For further detail, see the PAW OPERATOR'S HANDBOOK, chapter 'Revamping of VisualAge Pacbase Products'.

EXECUTION CONDITION

None.

ABNORMAL ENDING

Refer to Subchapter 'ABNORMAL EXECUTIONS' in Chapter: THE BATCH PROCEDURES.

If an abnormal end occurs, the procedure can be restarted with no additional modifications after 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 which are 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

EXTRACTION FOR PAW WORK STATIONS: PDSXTH

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

```
.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Input transaction file:
 -User check
  PACDMB  :    DSN=&&PACDMB

.Output file:
 -Extracted tables
  PACDMV

.Output report:
 -Extraction report
  PACDRH

.Sort files:
 SORTWK01, 02, 03
```

12.4. PROCEDURE JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - EXTRACTION OF TABLES FOR EXTERNAL LISTS -     -    *
//*********************************************************************
//$RADP.DEXH PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORTLIB
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    SYSTEM VSAM CATALOG
//  CYL=3,                 SORTWORK SIZE
//  SPAMV='(TRK,(150,15),RLSE)', SPACE OF EXTRACTED TRANSACTIONS
//  OUT='$OUT',            UTILITIES OUTPUT CLASS
//  OUTL='$OUT',           OUTPUT CLASS OF REPORTS
//  UWK=$UWK               WORK UNIT
//*********************************************************************
//* INPUT                                                             *
//*  COL 02     : *                                                   *
//*  COL 03     : DSMS USER CODE                                      *
//*  COL 11     : PASWORD                                             *
//*  COL 19-21  : PRODUCT CODE                                        *
//*  COL 22-23  : SUBSIDIARY CODE                                     *
//*  COL 24     : LANGUAGE CODE                                       *
//*                                                                   *
//*********************************************************************
//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=3200)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDSXTH   EXEC PGM=PDSXTH
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDMB   DD DSN=&&PACDMB,DISP=(OLD,DELETE)
//PACDMV   DD DSN=&&PACDIM,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAMV,
//            DCB=(RECFM=FB,LRECL=100,BLKSIZE=6200)
//PACDRH   DD SYSOUT=&OUTL
//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
```

13. BATCH UPDATE OF ENTITIES (DUPT)

13.1. INTRODUCTION

DUPT: INTRODUCTION

The purpose of the Batch Update of Entities procedure (DUPT) is to update 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" appendix of the DSMS Reference Manual.

EXECUTION CONDITION

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

ABNORMAL EXECUTION

Refer to Subchapter 'ABNORMAL EXECUTIONS' in Chapter: THE BATCH PROCEDURES.

Whatever the cause, the procedure can be restarted as it is once the problem is 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 concurrently. In this case, each Change or Event being created must be allocated a different temporary number. For exemple, 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 This field defines the update mode processing to be used by ALL userids for this execution of the DSMS batch procedure.
		A	NORMAL UPDATE MODE - Transactions are sorted in ascending order before any updates are applied (i.e. entity definitions are processed before sub-screen records.) - The upate mode is specified for each sign-on record.
		D	DELETE MODE - Transactions are sorted in descending order before any updates are applied. - All transactions are processed as deletes - Action Code D'. - Sign-on records must specify 'NORMAL' mode - all other modes will be considered an error.
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 processed 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 userid
11	8	...	DSMS userid 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 ERRORS INDICATOR
	blank	A blank line will be printed after each error message on the report.
	N	Blank lines will not be printed after error messages on the report.
35 1		REPORT PAGE BREAK INDICATOR
	blank	A new page will begin when the maximum number of lines per page has been exceeded.
	T	A new page will begin for each new transaction type.
	E	A page break will occur for each entity transaction type.
36 1		TRANSACTION SORT INDICATOR
	blank	The transactions will be sorted by type before they are processed.
	N	The transactions will be 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

UPDATE OF THE DSMS DATABASE: PDSUP0

```
.Permanent input-output files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Va Pac-element file
  PACDDC  :    DSN=&INDUV..&ROOTD.&ROOT2.0DC
 -Cross-reference file
  PACDDX  :    DSN=&INDUV..&ROOTD.&ROOT2.0DX

.Permanent input file:
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Input transaction file:
 -Update transactions obtained via
  the DEXP procedure
  PACDIM  :    DSN=&INPUT

.Output file:
 -Journal file
  PACDDJ  :    DSN=&INDUV..&ROOTD.&ROOT2.0DJ

.Output report:
 -Update review
  PACDRP

.Return codes:
 - 0:  No error
 -08:  Error on the user line code or on a parameter
 -12:  I/O error on a file.
```

13.4. EXECUTION JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - UPDATE OF THE DSMS DATABASE -                      *
//*********************************************************************
//$RADP.DUPT PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//  UWK='$UWK',            UNIT OF WORK FILES
//  CYL=3,                 SORTWORK SPACE
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  INPUT='NULLFILE'       INPUT TRANSACTIONS DSN
//*********************************************************************
//* INPUT :                                                           *
//* .. PARAMETERS LINE (OPTIONAL)                                     *
//*  COL 02     : $                                                   *
//*  COL 03     : UPDATE MODE (A,D)                                   *
//*  COL 04     : REPORT FORMAT INDICATOR (1,2,3)                     *
//* .. IDENTIFICATTION LINE (MANDATORY)                               *
//*  COL 01     : ACTION CODE / UPDATE MODE (V,R, )                   *
//*  COL 02     : *                                                   *
//*  COL 03-10  : USER CODE                                           *
//*  COL 11-18  : PASSWORD                                            *
//*  COL 19-21  : PRODUCT CODE                                        *
//*  COL 22-23  : SUBSIDIARY CODE                                     *
//*  COL 24     : (NOT USED)                                          *
//*  COL 25-31  : EXTERNAL REFERENCE VALUE (DATABASE, PRODUCT,        *
//*               SUBSIDIARY)                                         *
//*  COL 34     : BLANK LINE AFTER ERROR ( ,N)                        *
//*  COL 35     : REPORT PAGE BREAK INDICATOR ( ,T,E)                 *
//*  COL 36     : TRANSACTION SORT INDICATOR ( ,N)                    *
//*                                                                   *
//* .. COMMAND LINES                                                  *
//*********************************************************************
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDJ),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDX),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDSBAS EXEC PGM=PDSBAS
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PDSUP0 EXEC PGM=PDSUP0,REGION=4096K,COND=(0,NE,PDSBAS)
//******************************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
```

```
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDIM   DD DSN=&INPUT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRP   DD SYSOUT=&OUT
//SYSIN    DD DUMMY,DCB=BLKSIZE=80
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
```

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 CONDITION

None.

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 VA Pac DSMS components must have been executed.

ABNORMAL EXECUTION

Refer to Subchapter 'ABNORMAL EXECUTIONS' in Chapter: THE BATCH PROCEDURES.

If an abnormal end occurs, the procedure can be restarted with no additional modifications 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 requested 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 Languages 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 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 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 subsidi- ary codes and labels.

. In the User Parameters Tables (CHOICES: 'TUD', 'TUG', 'TUP' and 'TUS'): user codes and authorizations.

(For more details on 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

```
INITIAL DATABASE BACKUP:  PDSINI

.Input transaction file:
 -Initialization transaction
  PACDMB  :    DSN=&&DINIMB

.Permanent input file:
 -Error messages
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.Output file:
 -Sequential images of files
  PACDBB  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1)

.Output file:
 -Backup report
  PACDRU
```

14.4. EXECUTION JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - INITIALIZATION OF THE DSMS DATABASE -              *
//*********************************************************************
//$RADP.DINI PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',     3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',     INDEX OF USER   NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',    LIBRARY OF LOAD-MODULES
//  DSCB='$DSCB',       DSCB MODEL FILE
//  OUT='$OUT',         OUTPUT CLASS
//  VOLS='SER=$VOLUN',  VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',    BACKUP UNIT (DISK OR TAPE)
//  UWK='$UWK',         WORKFILE UNIT
//  SPABB='(TRK,(10,2),RLSE)' SPACE OF BACKUP (IF DISK)
//*********************************************************************
//*  INPUT                                                            *
//*  COL  2     : I                                                   *
//*  COL  3     : INITIAL LANGUAGE CODE ( F=FRENCH, E=ENGLISH)        *
//*  COL  4     : 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=3200)
//PDSINI EXEC PGM=PDSINI
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDMB   DD DSN=&&DINIMB,DISP=(OLD,PASS)
//PACDBB   DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPABB,
//         DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDRU   DD SYSOUT=&OUT
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT
```

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 EXECUTIONS'.

Whatever may be the reason for the abnormal end, the procedure can be reexecuted 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 ! User code                          !
! 11 !   8 ! pppppppp ! User password                      !
!----+-----+----------+------------------------------------!
!                   Optional                               !
!----+-----+----------+------------------------------------!
! 19 !   3 !   ppp    ! Product code                       !
! 22 !   2 !   su     ! Subsidiary code                    !
! 24 !   1 !'F' ou '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, on demand, the list of formatted transactions.

RESULT

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

15.3. DESCRIPTION OF STEPS

```
TRANSACTION EXTRACTION AND FORMATTING:  PDS700

.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE
 -Archived DSMS journal
  PACDBJ  :    DSN=&INDUN..&ROOTD.&ROOT2.0BJ

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

.Output file:
 -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 on a parameter
       The environment definition is missing.
 -12:  File access error
       The technical record is missing.
```

15.4. EXECUTION JCL

```
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*              - EXTRACTION OF DSMS JOURNAL -                        *
//**********************************************************************
//$RADP.DXBJ PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//  SPAIM='(TRK,(15,5))',   SPACE OF EXTRACTED TRANSACTIONS
//  OUT='$OUT',            OUTPUT CLASS
//  UWK='$UWK'             WORK UNIT
//**********************************************************************
//*   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                                            *
//*                                                                    *
//**********************************************************************
//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=6160)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDS700 EXEC PGM=PDS700
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,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=&INDUN..&ROOTD.&ROOT2.0BJ(0),DISP=SHR
//PACDRK   DD SYSOUT=&OUT
//PACDSK   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

16. CODE OR 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 REQUIREMENT

This procedure works from a sequential backup, it must therefore be preceded by a backup.

ABNORMAL EXECUTION

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

Whatever the reason of the abnormal end, the procedure can be executed again 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 subsidary code 'ss'
		'**'	Changes made on the entities which depend on all the subsidary codes
24	1	'E' or 'F'	Language code
COMPULSORY: 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 l 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, subsidaries or sites new codes must not already exist (in the same subsidary 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 impredictable.

- 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 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 modification was performed on the archive (1 in column 26): new version of the Journal's sequential backup.

If the modification was performed on the Database backup (1 in column 25): 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  ! Invalid absence of backup/archive 'tops'        !
    ! 12  ! Input/Output error or inconsistent DSMS base    !
    ! 16  ! Sort error                                      !
    +-------------------------------------------------------+
```

16.3. DESCRIPTION OF STEPS

DREN: DESCRIPTION OF STEPS

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

```
UPDATES:  PDSMS

.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Error messages
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE
 -Cross-references
  PACDDX  :    DSN=&INDUV..&ROOTD.&ROOT2.0DX
 -DSMS backup
  PACDBB  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(0)
 -DSMS archiving
  PACDBJ  :    DSN=&INDUN..&ROOTD.&ROOT2.0BJ(0)

.Input file:
 -User queries
  PACDMB  :    DSN=&&PACDMB

.Work files:
 -Update requests
  PACDW0
 -Partial backup (sorted)
  PACDW1
 -Partial backup (non 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

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - CHANGE OF TABLE AND SITE CODES, AND KEYWORDS -     *
//*********************************************************************
//$RADP.DREN PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',       SORT LIBRARY
//  SPAWK='(TRK,(60,5),RLSE)',  WORK SPACE (DA + DC)
//  UWK=$UWK,              UNIT OF WORK FILES
//  LNG=E,                 LANGUAGE OF MONITOR (E FOR ENGLISH)
//  CYL=3,                 SORTWORK SPACE
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  DSCB='$DSCB',          DSCB MODEL FILE
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  SPABB='(TRK,(10,2),RLSE)',  SPACE OF BACKUP (IF DISK)
//  SPABJ='(TRK,(10,2),RLSE)'   SPACE OF BACKUP (IF DISK)
//*********************************************************************
//* INPUT :                                                           *
//* .. IDENTIFICATION LINE                                            *
//*  COL 02     : *                                                   *
//*  COL 03     : DSMS USER CODE                                      *
//*  COL 11     : PASSWORD                                            *
//*  COL 19-21  : PRODUCT CODE OR '***'                               *
//*  COL 22-23  : SUBSIDIARY CODE OR '**'                             *
//*  COL 24     : LANGUAGE CODE                                       *
//*  COL 25     : MODIFICATIONS ON SAVE (1, )                         *
//*  COL 26     : MODIFICATIONS ON ARCHIVE (1, )                      *
//* ..  MODIFICATION(S) COMMAND LINE(S)                               *
//*  COL 02-04  : TYPE OF MODIFICATION                                *
//*  COL 05-17  : OLD CODE                                            *
//*  COL 18-30  : NEW CODE                                            *
//*                                                                   *
//*********************************************************************
//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=3200)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDSMS  EXEC PGM=PDSMS&LNG,REGION=4096K
//***********************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDMB   DD DSN=&&PACDMB,DISP=SHR
//PACDBB   DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(0),DISP=OLD
//PACDBJ   DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(0),DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
```

```
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,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=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//            DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDJB   DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPABJ,
//            DCB=(&DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//PACDIA   DD SYSOUT=&OUT
//PACDIK   DD SYSOUT=&OUT
//PACDJK   DD SYSOUT=&OUT
//PACDKK   DD SYSOUT=&OUT
//PACDLK   DD SYSOUT=&OUT
//PACDIS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//DEL12  EXEC PGM=IEFBR14,COND=(12,NE,PDSMS)
//******************************************
//DDBB     DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),DISP=(OLD,DELETE)
//DDBJ     DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(+1),DISP=(OLD,DELETE)
```

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

The COBOL sources of the programs that contain DAF operators DAF 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	blanc	(Not used)
23	5	nnnns	Session number - Session status
28	1	blanc	(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	blanc	(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 a list of errors, if applicable.

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 unique routine which acts as a flow monitor for all the programs, considered as sub-programs of this monitor. It includes the following steps:

GENERATED PROGRAM'S PRE-PROCESSOR: DAFD10

```
.Permanent input files:
 -Data file
  PACDDA  :    DSN=&INDUV..&ROOTD.&ROOT2.0DA
 -Error message file
  PACDDE  :    DSN=&INDSV..&ROOTD.00DE

.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

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - ACCESS FACILITY PRE-PROCESSING -                   *
//*********************************************************************
//$RADP.DPDF PROC ROOTD=$ROOTD,   2 FIRST CHARACTERS TRANSACTION
//         ROOT2=$ROOT2,          3RD CHARACTER TRANSACTION
//         INDSV='$INDSV',      INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',      INDEX OF SYSTEM NON VSAM FILES
//         INDUV='$INDUV',      INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',       USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',       SYSTEM VSAM CATALOG
//         STEPLIB='$MODB',   LIBRARY OF LOAD-MODULES
//         OUT=$OUT,              OUTPUT CLASS
//         SPAMB='(TRK,(150,15))', SPACE OF GENERATED PROGRAMS
//         UWK=$UWK               WORK UNIT
//***************************************************************
//INPUT  EXEC PGM=PTU001
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&DPDFMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//DAFD10 EXEC PGM=DAFD10
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//DAF80    DD DSN=&&DPDFMB,DISP=(OLD,DELETE)
//COB80    DD DSN=&&DAFGEN,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//DAFREP   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

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 EXECUTIONS

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

```
TRANSACTION FORMATTING:  PDS900

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

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

18.4. PROCEDURE JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                              *
//*                                                                     *
//*              - BATCH UPDATE FROM DAF TABLES -
//***************************************************************
//$RADP.DUPT PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTIONT
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDSN='$INDSN',        INDEX OF SYSTEM NON VSAM FILES
//  INDUV='$INDUV',        INDEX OF USER VSAM FILES
//  INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILE
//  DAFINPUT=,             INPUT TRANSACTIONS DSN
//  STEPLIB='$MODB',       UNIT OF WORK FILES
//  UWK='$UWK',            SORTWORK SPACE
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  OUT='$OUT',            OUTPUT CLASS
//  SPAIM='(TRK,(100,10),RLSE)'   TRANSACTION SPACE
//***************************************************************
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDA),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDC),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDJ),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDX),DISP=SHR
//         DD DSN=&INDSN..&ROOTD.&ROOTD.SY(VERIFDE),DISP=SHR
//PDSBAS EXEC PGM=PDSBAS
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRS   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP 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=PDSUP0,REGION=4096K,COND=(0,NE,PDSBAS)
//******************************************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//PACDIM   DD DSN=&&PACDIM,DISP=(OLD,DELETE)
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDJ   DD DSN=&INDUV..&ROOTD.&ROOT2.0DJ,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDRP   DD SYSOUT=&OUT
```

```
//SYSIN    DD DUMMY,DCB=BLKSIZE=80
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

19. INSTALLATION

19.1. INTRODUCTION

GENERAL INTRODUCTION

The installation procedure is broken down into three main phases:

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

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

Before proceeding to the actual installation, the user must be familiar with the technical characteristics of the VisualAge Pacbase DSMS function described in this manual. This information is needed to prepare the required environment for the installation procedure (disk space, various catalogues, 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.

19.2. INSTALLATION CARTRIDGE / TAPE

INSTALLATION TAPE

The installation tape or "3480" cartridge (6,250 BPI, standard labels) contains the following files:

RANK	LABEL	LRECL	BLKS	CONTENTS
1	INST.JCL	80	11440	Initial preparation JCL.
2	INST.MOD		6144	MM1JCL Load-module, utility for JCL preparation.
3	PACD.JCL	80	11440	JCL skeleton for installation and operation of DSMS.
4	PAC.DE	90	6300	System file of DSMS error messages.
5	PACD.MBR8		6144	Batch load-modules.
6	PACD.MTR8		6144	On-Line load-modules.
Test deck : ---------				
7	PAC.BB	354 (VB)	6376	Backup file for the DSMS test database.
8	PACD.SOURCE	80	6080	Used ctl sources and DAF dictionary
For correction tapes or cartridges				
9	PACD.PATCH	80	6080	Patch JCL PDS
10	PACD.README	133 (FBA)	5320	README file Info. on correction release

19.3. INSTALLATION PREPARATION

PREPARATION

It is recommended to copy all the VA Pac-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 3390 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 cardridge's initial JCL ('INST.JCL') by one of the site's utilities (i.e. IEBGENER), in the PDS mentioned above.

USE

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 cardridge (or tape) provided by IBM onto a user cardridge (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 specialy 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 cardridge 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, either a sequential file
                          chosen by the user.
```

Enter the parameters (see following sub-chapters).

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

19.3.1. INITIAL JCL

```
$CO****    VA PACDSMS 2.5 MVS/CICS $VV $DATE    ************
//PACDSMS0 JOB (---),'3480',CLASS=D,MSGCLASS=A
//ALLOC   EXEC PGM=IEHINITT
//TCGI     DD  DISP=SHR,UNIT=(3480,,DEFER),VOL=(,RETAIN,SER=$CGICAR)
//TINST    DD  DISP=SHR,UNIT=(3480,,DEFER),VOL=(,RETAIN,SER=------)
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
TINST INITT SER=------,OWNER='------',DISP=REWIND
/*
//PACCOP   PROC INDEX='$INPRO',NAME=XXX,LAB=N
//GENER   EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  DUMMY
//SYSUT1   DD  DSN=&INDEX..&NAME,DISP=SHR,
//             VOL=(,RETAIN,REF=*.ALLOC.TCGI),LABEL=&LAB
//SYSUT2   DD  DSN=&INDEX..&NAME,DISP=(,KEEP),
//             VOL=(,RETAIN,REF=*.ALLOC.TINST),LABEL=&LAB,
//             DCB=*.SYSUT1
//        PEND
//STEP1   EXEC PACCOP,LAB=01,NAME=JCL,INDEX=INST
//STEP2   EXEC PACCOP,LAB=02,NAME=MOD,INDEX=INST
//STEP3   EXEC PACCOP,LAB=03,NAME=JCL
//STEP4   EXEC PACCOP,LAB=04,NAME=DE,INDEX=PAC
//STEP5   EXEC PACCOP,LAB=05,NAME=MBR8
//STEP6   EXEC PACCOP,LAB=06,NAME=MTR8
//STEP7   EXEC PACCOP,LAB=07,NAME=BB,INDEX=PAC
//STEP8   EXEC PACCOP,LAB=08,NAME=SOURCE
//STEP9   EXEC PACCOP,LAB=09,NAME=PATCH
//STEP10  EXEC PACCOP,LAB=10,NAME=README
//PACDSMS1  JOB (---),'BLDX MM1JCL',CLASS=D,MSGCLASS=A
//COPY    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=2
//OM       DD DISP=SHR,DSN=---.---.---
//SYSIN    DD *
 C  I=((IM,R)),O=OM
/*
//PACDSMS2  JOB (---),'JCL INSTALLATION',CLASS=D,MSGCLASS=A
//MM1JCL EXEC PGM=MM1JCL
//STEPLIB  DD DISP=SHR,DSN=---.---.---
//SYSOUT   DD SYSOUT=A
//SYSUT1   DD DSN=PACD.JCL,DISP=OLD,
//            UNIT=3480,VOL=(,RETAIN,SER=------),LABEL=3
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT8   DD DUMMY,DCB=BLKSIZE=1370
//SYSUT9   DD DUMMY,DCB=BLKSIZE=1370
//***********************************************************************
//*  --- CREATION DU JCL D'INSTALLATION PAR L'UTILITAIRE 'MM1JCL' --- *
//*      ********************************************************       *
//*                                                                     *
//*      .MODIFIER LA LISTE DES COMMANDES FOURNIES EN PRECISANT LES     *
//*       VARIANTES D'INSTALLATION (S'IL Y EN A), EN DEMANDANT  SI      *
//*       NECESSAIRE LA SELECTION DE PORTIONS DE JCL D'INSTALLATION     *
//*       (MODULES DE JCL), EN DONNANT DES VALEURS APPROPRIEES AUX      *
//*       PARAMETRES D'INSTALLATION, ET EN PRECISANT EVENTUELLEMENT     *
//*       DES LIGNES A AJOUTER EN TETE OU EN FIN DE CHAQUE MODULE       *
//*       DE JCL.                                                       *
//***********************************************************************
//*                                                                     *
//*  --- CREATION OF INSTALLATION JCL THROUGH PROGRAM 'MM1JCL' ---      *
//*      *****************************************************          *
//*                                                                     *
//*      .MODIFY THE LIST OF THE SUPPLIED COMMANDS BY ENTERING THE      *
//*       INSTALLATION VARIANTS (IF ANY), BY ASKING, IF NECESSARY, A    *
//*       SELECTION OF PARTS OF INSTALLATION JCL (JCL MODULES), BY      *
//*       GIVING THE APPROPRIATE VALUES TO THE INSTALLATION PARAMETERS*
//*       AND BY SPECIYING (OPTIONAL) THE LINES TO BE ADDED AT THE    *
//*       BEGINNING OR AT THE END OF EACH JCL MODULE.                 *
//*********************************************************************
//SYSPRM   DD DUMMY
//SYSUT2   DD ---------   MEMBRE DE PDS OU FICHIER SEQUENTIEL
//*                       RECEVANT LE JCL (LRECL=80)
//*                       PDS MEMBER OR SEQUENTIAL FILE RECEIVING
//*                       THE INSTALLATION JCL (LRECL=80)
//SYSIN DD *
===SELL <>                .LANGUAGE SELECTION (ENG OR FRA)
===SELV DAF               .DSMS ACCESS FACILITY
                          .MODULE DAF (DSMS ACCESS FACILITY)
===SELV ROLD              .SELECTED PROC FOR OLD DSMS RETRIEVAL
                          .SELECTION PROC REPRISE ANCIEN DSMS
===SELV SEC               .SECURITY SYSTEM INTERFACE
                          .INTERFACE SYSTEME DE SECURITE
===PRM PRFJ=PAC           .JOB NAMES PREFIXES (3 CHARACTERS)
                          .PREFIXE DES NOMS DE JOBS (3 CARACTERES
===PRM CCPT=<>            .JOB ACCOUNTING CODES (JOB CARDS)
                          .CODE COMPTABLE DES JOBS (CARTES JOB)
===PRM CLASSJ=1           .JOB EXECUTION CLASS (JOB CARDS)
                          .CLASSE D'EXECUTION DES JOBS (CARTES JO
===PRM MSGCL=A            .JCL OUTPUT CLASS (MSGCLASS)
                          .CLASSE DE SORTIE DU JCL (MSGCLASS)
===PRM U3480=3480         .CARTRIDGE UNIT
                          .UNIT CARTOUCHE
===PRM ICART=<>           .INSTALLATION CARTRIDGE VOL=SER=
                          .VOL=SER= CARTOUCHE INSTALLATION
===PRM OUT=A              .JOB SYSOUT CLASS
                          .CLASSE DE SYSOUT DANS LES JOBS
===PRM INDSV='EXP.DSM25'  .DSMS SYSTEM VSAM FILES INDEXES
                          .INDEX FICHIERS DSMS VSAM
===PRM INDUV='CICS.DSM25' .DSMS USER VSAM FILES INDEXES
                          .INDEX FICHIERS UTILISATEUR DSMS VSAM
===PRM INDSN='EXP.DSM25'  .DSMS SYST.NON VSAM FILES INDEXES
                          .INDEX FICHIERS DSMS NON VSAM
===PRM INDUN='PDS.R25'    .DSMS USER NON VSAM FILES INDEXES
                          .INDEX FICHIERS UTILISATEUR DSMS NON VS
===PRM ROOTD='DS'         .ROOT OF DSMS SYSTEM (2 CHARACTERS)
                          .RADICAL DSMS(2 CARACTERES)
===PRM ROOT2='M'          .3RD CHARACTER OF DSMS TRANSACTION
                          .3EME CARACTERE TRANSACTION DSMS
===PRM RADP='DS25'        .PREFIX OF CATALOGUED PROCEDURE NAMES
                          .PREFIXE DES NOMS DE PROCEDURES CATALOG
===PRM VOLSV=<>           .VOL.OF DSMS SYSTEM VSAM FILES
                          .VOLUME FICH. DSMS SYSTEME VSAM
===PRM VOLSN=<>           .VOL.OF DSMS SYSTEM NON VSAM FILES
                          .VOLUME FICH. DSMS SYSTEME NON VSAM
===PRM VOLUV=<>           .VOL.OF DSMS USER VSAM FILES
                          .VOLUME FICH. DSMS UTILISATEUR VSAM
===PRM VOLUN=<>           .VOL.OF DSMS USER NON VSAM FILES
                          .VOLUME FICH. DSMS UTILIS. NON VSAM
===PRM SCAT=<>            .CATALOG OF DSMS SYSTEM VSAM FILES
                          .CATALOGUE VSAM FICH. SYSTEME DSMS
===PRM VCAT=<>            .CATALOG OF DSMS USER VSAM FILES
                          .CATALOGUE VSAM FICH. UTILISATEURS DSMS
===PRM UWK=SYSDA          .WORK UNIT
                          .UNITE DE TRAVAIL
===PRM UNITSN=SYSDA       .NON VSAM DSMS SYSTEM FILES UNIT
                          .UNIT FICH.PERMANENTS DSMS NON VSAM
===PRM UNITUN=SYSDA       .NON VSAM DSMS USER FILES UNIT
                          .UNIT FICH.PERMANENTS UTILI. NON VSAM
===PRM MODB='PDS.R25.MBR8' .BATCH PACBASE DSMS 2.5 LIBRARY
                           .BIBLI.BATCH PACBASE DSMS 2.5
===PRM MODT='PDS.R25.MTR8' .TP PACBASE DSMS 2.5   LIBRARY
                           .BIBLI.TP PACBASE DSMS 2.5
===PRM BIBP='SYS1.PROCLIB' .PROCEDURE LIBRARY
                           .BIBLIOTHEQUE DES PROCEDURES
===PRM BIBT='SYS1.SORTLIB' .SORT LIBRARY
                           .BIBLIOTHEQUE DE  TRI
===PRM DSCB='PDS.DSCB'     .DSCB MODEL FILE DSNAME
                           .DSNAME FICHIER DSCB MODELE
```

```
===PRM CSDL='CICS330.LOADLIB'  .DFHCSDUP STEPLIB DSN
                               .DSN STEPLIB DFHCSDUP
===PRM DFHCSD='PAC.DFHCSD'     .DSMS CSD DSN
                               .DSN CSD DSMS
===PRM LIST=<>                 ."LIST" NAME
                               .NOM DE "LIST"
===PRM GROUP='PACDSMS'         ."GROUP" NAME FOR PROGRAMS
                               .NOM DE "GROUP" POUR PROGRAMMES
===PRM GROUT='PACDSMS'         ."GROUP" NAME FOR TRANSACTION
                               .NOM DE "GROUP" POUR TRANSACTION
===PRM GROUF='PACDSMS'         ."GROUP" NAME FOR FILES
                               .NOM DE "GROUP" POUR FICHIERS
===BEGMOD
./ ADD NAME=$MODULE
/*
//
```

19.3.2. INSTALLATION OF COMPLETE JCL

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 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 before/after lines
```

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.

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 is OK 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' Cisize/Blksize' (VA Pac only).

CAUTION:

. 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.

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 IEBUPDTE were replaced with ':/'.

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

REPORT

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

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 '$' in the flow processed by MM1JCL which are NOT parameters (in particular, PACDESIGN transactions).

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

NOTE: 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         )
```

19.3.3. 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.

.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>
```

19.3.4. 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

19.3.5. JCL MODULES

TABLE OF JCL MODULES: ===SELM mmmm1 mmmm2 ... mmmmn

mmmm	CONTENTS	NATURE
	CICS UPDATE:	
DCICSD	Update CSD	JCL
	DSMS INSTALLATION:	
DI1SY	Allocation/parameters PDS loading	-
DI2PRE	GDG allocation/backup file initit.	-
DI3PGM	Allocation/load modules loading	-
DI4DE	DE file installation	-
DI5PRO	Cataloging operations procedures	-
	TEST DECK FILE LOADING:	
DI6DS	Load test deck file (DRST)	-
DI7SRC	Allocation/source PDS loading for user checks	-
DI8SER	RACF security system complement	-

TABLE OF JCL MODULES (cont'd)

mmmm	CONTENTS	NATURE
	RETRIEVAL OF THE DSMS 8.0.1 DATABASE	
DI7R801	Retrieval of BB 8.0.1 backup (DR80)	JCL
	RETRIEVAL OF THE DSMS 8.0.2 DATABASE COMPATIBLE WITH PACBASE 8.0.1 WHEN MIGRATING FROM PACBASE 8.0.1 TO PACBASE 8.0.2.	
DRPB802	8.0.2 programs installation and 8.0.2 DC file definition.	JCL
	Retrieval of BB 8.0.2 backup (DR8X)	MVS JCL
	RETRIEVAL OF THE DSMS 1.5 WHEN MIGRATING FROM 8.0.2 DATABASE	
DI7R802	Retrieval of BB 8.0.2 backup (DR8Q)	JCL
	RETRIEVAL OF A 1.2 OR 1.5 DSMS DATABASE IN THE 2.5 RELEASE	
DI7R150	Retrieval of BB 1.2/1.5 backup (DR15)	JCL
DI7R15J	Retrieval of BJ 1.5 backup (DR5J)	JCL

TABLE OF JCL MODULES (cont'd)

mmmm	CONTENTS	NATURE
	EXAMPLES OF TEST JOBS	
JCLDARC	Journal archiving	JCL
JCLDEXT	Extraction (DEXT) and possible update (DUPT)	-
JCLDRST	Database Restoration	-
JCLDINS	List of program dates	-
JCLDSAV	Database Backup	-
JCLDXBJ	Journal extraction in the form of update transactions	-
JCLDAF	Execution of a batch program for DSMS access via DAF	-

19.3.6. JCL PARAMETERIZATION

TABLE OF PARAMETERS

```
===PRM PPPP=pppp                .Comments

+-------+---------------------------------------+-----------+
!CODE   ! MEANING                               ! DEFAULT   !
!PPPP   !                                       ! pppp      !
+-------+---------------------------------------+-----------+
!       !                                       !           !
!       ! ON THE JOB CARDS                      !           !
!       ! ----------------                      !           !
!       !                                       !           !
!PRFJ   ! 3-character job name prefix           ! PAC       !
!CCPT   ! Job accounting code                   ! <>        !
!CLASSJ ! Job execution class                   ! 1         !
!MSGCL  ! JCL output class                      ! A         !
!       !                                       !           !
!       ! CODING OF DSN's                       !           !
!       ! ---------------                       !           !
!       !                                       !           !
!       ! All DSMS files (except load-module    !           !
!       ! libraries) are named as follows:      !           !
!       !                                       !           !
!       ! INDUV.xxy0ff : User VSAM file         !           !
!       ! INDUN.xxy0ff : User non-VSAM file     !           !
!       ! INDSV.xx00DE : Error message VSAM f.  !           !
!       ! INDSV.xxy0DF : DAF VSAM               !           !
!       ! INDSN.xxxxSY : PDS 'SY' non-VSAM file!           !
!       !                                       !           !
!       ! IND--  Index of file names:           !           !
!INDSV  ! VSAM system                           ! EXP.PAC80 !
!INDSN  ! Non-VSAM system (SAM, PDS)            ! EXP.PAC80 !
!INDUV  ! VSAM user                             ! CICS.PAC80!
!INDUN  ! Non-VSAM user (SAM)                   ! PAC.PAC80 !
!       !                                       !           !
!       ! xx=ROOTD, y=ROOT2, ff=file code       !           !
!ROOTD  ! System root (2 characters)            ! DS        !
!ROOT2  ! DSMS installation sub-code (1 char-   ! M         !
!       ! acter)                                !           !
!       !                                       !           !
+-------+---------------------------------------+-----------+
```

TABLE OF PARAMETERS (Cont'd.)

CODE	MEANING	DEFAULT
PPPP		pppp
	ON THE DD CARDS	

OUT	SYSOUT class	A
U3480	UNIT installation cardridge copy of cardridge provided by IBM.	3480
ICART	VOL=SER installation cardridge, copy of cardridge provided by IBM.	<>
UWK	UNIT of work files used	SYSDA
UNITSN	UNIT of non-VSAM system files	SYSDA
UNITUN	UNIT of non-VSAM user files	SYSDA
VOLSN	Volume name of non-VSAM system files	<>
VOLSV	Volume name of VSAM system files	<>
VOLUN	Volume name of non-VSAM user files	<>
VOLUV	Volume name of VSAM user files	<>
	CICS CSD UPDATE PARAMETERS (CICS/ESA V3 RDO only)	
CSDL	DFHCSDUP STEPLIB DSN	'CICS311.LOADLIB'
DFHCSD	VA Pac CSD DSN for update	'PAC.DFHCSD'
GROUP	Program CSD input group	PACDSMS
GROUT	Transaction CSD input group	PACDSMS
GROUF	File CSD input group	PACDSMS
LIST	List in which the DSMS group is to be added	<>

TABLE OF PARAMETERS (Cont'd.)

CODE PPPP	MEANING	DEFAULT pppp
	OTHER PARAMETERS ----------------	
RADP	Prefix of names of procedures to install (4 characters maximum).	DSMS
VCAT	DSNAME of User Files VSAM Catalog	
SCAT	DSNAME of System Files VSAM Catalog	
MODB	DSNAME of DSMS batch load module library.	
MODT	DSNAME of DSMS on-line load module library.	
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
DSCB	DSNAME of the DSCB MODEL file.	PDS.DSCB

Note: The '<>' indicates a required parameter.

VERY IMPORTANT:

- Values of parameters containing special characters must be delimited by quotes.

- Comments on ===PRM lines should not exceed column 72. They should be preceded by a '.'.

19.3.7. JCL MODULE SEPARATORS

JCL BEFORE/AFTER LINES

```
===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 adds ....1 to ....n lines in front of each JCL module and ....1 to ....n lines to the end of each JCL module.

19.4. INSTALLATION STEPS

INSTALLATION PROCESS

Once all JCL is obtained, the installation of the DSMS system is broken down into ten phases:

1. Update of CICS,
2. Allocation/Loading of the SY parameters PDS,
3. GDG allocation/Initialization of backup files,
4. Allocation/Loading of the load-module libraries,
5. Definition/Loading of the error message file/HELP function,
6. Cataloging of operations procedures,
7. Loading of the DSMS test deck files,
8. CICS modification,
9. Additional user operations.
10. Additional Security System Interface

19.4.1. CICS CSD UPDATE

```
1. UPDATE OF CICS

 .Transaction:
    1 user transaction code: xxyE
    You must declare the transaction xxyF if you also intend
    to work in French.

 .Programs:
    xx00nn for programs,
    (See complete list in Chapter 'DSMS
    COMPONENTS' Subchapter 'ON-LINE PROGRAM LIBRARY').

 .Files:

    2 DSMS files:

    xx00DE : Error messages/HELP function,
    xxy0DF : Work file for DSMS Access Facility (DAF)

    4 user files :

    xxy0DA : Data,
    xxy0DX : Cross-references,
    xxy0DC : VisuaAge Pacbase elements,
    xxy0DJ : Update Journal.
```

NOTES ON INSTALLATION:

Entries are updated in the CSD by the $prfj.0CI job of the ===MOD DCICSD JCL module.

PREVIOUS CICS RELEASES: entries for assembly are found in following JCL modules:

The option specifying the Dynamic Backout is compulsory for the DA, DC, DX and DJ files.

The CICS parameter 'SPOOL=YES' is compulsory to submit batch jobs on-line (choice : LVQ).

xx = first 2 characters of the transaction code ($rootd), rootd),

y = third character of the transaction code ($root2).
CAUTION: The value of 'y' must not be H.

```
//$PRFJ.0CI JOB ($CCPT),'PACDSMS DFHCSDUP',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
//**********************************************************************
//* VisualAge Pacbase-DSMS                                              *
//*                                                                     *
//*               - 'DFHCSD' BATCH UPDATE -                            *
//*********************************************************************
//DFHCSDUP EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=$CSDL,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DFHCSD   DD DSN=$DFHCSD,DISP=SHR
//SYSIN    DD *
***********************************************************************
*   CSD CICS : VA Pac     DSMS   $ROOTD COMMON FILE
***********************************************************************
DEFINE FILE($ROOTD.00DE) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS HELP)
       DSNAME($INDSV..$ROOTD.00DE)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(F)
       ADD(NO) BROWSE(YES) DELETE(NO) READ(YES) UPDATE(NO)
DEFINE FILE($ROOTD.$ROOT2.0DF) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS ACCESS FACILITY)
       DSNAME($INDSV..$ROOTD.$ROOT2.0DF)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
***********************************************************************
*   CSD CICS : VA Pac     DSMS   $ROOTD.$ROOT2 DATABASE FILES
******************************************************************
DEFINE FILE($ROOTD.$ROOT2.0DA) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS DATA)
       DSNAME($INDUV..$ROOTD.$ROOT2.0DA)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOTD.$ROOT2.0DC) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS PACBASE ELEMENTS)
       DSNAME($INDUV..$ROOTD.$ROOT2.0DC)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOTD.$ROOT2.0DX) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS XREF)
       DSNAME($INDUV..$ROOTD.$ROOT2.0DX)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3) INDEXBUFFERS(2)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($ROOTD.$ROOT2.0DJ) GROUP($GROUF)
DESCRIPTION(PACBASE DSMS JOURNAL)
       DSNAME($INDUV..$ROOTD.$ROOT2.0DJ)
       STRINGS(1)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(3)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
***********************************************************************
*   CSD CICS : VA Pac     DSMS   $ROOTD. PROGRAMS
```

```
    **********************************************************************
    DEFINE PROGRAM($ROOTD.CHOI) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.CUAM) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.CUEV) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.CUMQ) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.CURQ) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.CUSI) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.MONI) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.MOSO) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.TPDF) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00AA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00AB) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00BA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00B1) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00B2) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00B3) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00B4) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00B5) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00EA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00E1) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00E2) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00E3) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00E4) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00E5) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00E6) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00FA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00FB) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00HE) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00JO) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00KA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00K1) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00K2) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00K3) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00LE) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00LS) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00MA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00PA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00P1) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00QA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00QB) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00QC) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q1) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q2) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q3) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q4) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q5) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q6) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q7) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q8) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00Q9) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00SA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00SI) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S1) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S3) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S4) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S5) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S6) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S7) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S8) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00S9) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TA) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TT) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TU) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TV) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TW) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TX) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TY) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00TZ) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00T1) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00T2) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00T3) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00T4) GROUP($GROUP)
    DEFINE PROGRAM($ROOTD.00T5) GROUP($GROUP)
```

```
DEFINE PROGRAM($ROOTD.00T6) GROUP($GROUP)
DEFINE PROGRAM($ROOTD.00T7) GROUP($GROUP)
DEFINE PROGRAM($ROOTD.00T8) GROUP($GROUP)
DEFINE PROGRAM($ROOTD.00T9) GROUP($GROUP)
DEFINE PROGRAM($ROOTD.00UD) GROUP($GROUP)
DEFINE PROGRAM($ROOTD.R980) GROUP($GROUP)
DEFINE PROGRAM($ROOTD.UCTR) GROUP($GROUP)
                            RESIDENT(YES)
===SEQ FOR SEC
DEFINE PROGRAM($ROOTD.SECT) GROUP($GROUP)
===SEQ
**********************************************************************
*   CSD CICS : VA Pac  DSMS   $ROOTD.$ROOT2.E TRANSACTION
**********************************************************************
DEFINE TRANSACTION($ROOTD.$ROOT2.E) GROUP($GROUT)
DESCRIPTION(PACBASE DSMS TRANSACTION) PROGRAM($ROOTD.MONI)
ADD GROUP($GROUP) LIST($LIST)
ADD GROUP($GROUF) LIST($LIST)
ADD GROUP($GROUT) LIST($LIST)
//
```

19.4.2. SYSTEM PARAMETERS PDS ALLOCATION/LOADING

2. ALLOCATION/LOADING OF THE SY PARAMETERS PDS

JOB $prfj.1SY

CAUTION: change :/ characters with ./ before executing the job.

DELETE: IDCAMS : parameter file deletion,

BR14 : IEFBR14 : parameter file allocation,

UPD : IEBUPDTE : file member loading.

See the description in Subchapter 'SYSTEM PARAMETER LIBRARY', Chapter: DSMS COMPONENTS.

```
//$PRFJ.1SY  JOB ($CCPT),'SY PDS',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*                                                                   *
//*    ALLOCATION OF VA PAC DSMS PARAMETER PDS                        *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*                                                                   *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*  - REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*  - THIS PDS CONTAINS THE SYSINS FOR ALLOCATING THE FILES THAT     *
//*  MAKE UP THE PACBASE DSMS FILES                                   *
//*  THE INDICATED SIZES CAN BE ADAPTED ACCORDING TO YOUR NEEDS       *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOTD.$ROOTD.SY)
//*
//STEP2    EXEC PGM=IEFBR14
//SY       DD DSN=$INDSN..$ROOTD.$ROOTD.SY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),VOL=SER=$VOLSN,
//         SPACE=(6080,(10,,5)),UNIT=$UNITSN
//*
//********************************************************************
//*        --- LOADING OF VA PAC DSMS PARAMETERS IN 'SY' PDS         *
//********************************************************************
//UPD    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDSN..$ROOTD.$ROOTD.SY,DISP=SHR
//SYSIN    DD *
:/     ADD NAME=DF$ROOTD.$ROOT2.0DA
 DELETE ($INDUV..$ROOTD.$ROOT2.0DA) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOTD.$ROOT2.0DA) -
                  SHR (2,3) REUSE  KEYS (40,2)        -
                  VOL ($VOLUV)  CYL (5,1)           -
                  RECSZ (80,350) )                   -
        INDEX   ( NAME ($INDUV..$ROOTD.$ROOT2.0DA.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDUV..$ROOTD.$ROOT2.0DA.D) -
                  FSPC (10,5)                         -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$ROOTD.$ROOT2.0DC
 DELETE ($INDUV..$ROOTD.$ROOT2.0DC) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOTD.$ROOT2.0DC) -
                  SHR (2,3) REUSE  KEYS (31,2)        -
                  VOL ($VOLUV)  CYL (1,1)           -
                  RECSZ (50,168) )                   -
        INDEX   ( NAME ($INDUV..$ROOTD.$ROOT2.0DC.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDUV..$ROOTD.$ROOT2.0DC.D) -
                  FSPC (10,5)                         -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$ROOTD.$ROOT2.0DX
 DELETE ($INDUV..$ROOTD.$ROOT2.0DX) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOTD.$ROOT2.0DX) -
                  SHR (2,3) REUSE  KEYS (50,0)        -
                  VOL ($VOLUV)  CYL (5,1)           -
                  RECSZ (80,80) )                   -
        INDEX   ( NAME ($INDUV..$ROOTD.$ROOT2.0DX.I) -
                  CISZ (1096) )                       -
        DATA    ( NAME ($INDUV..$ROOTD.$ROOT2.0DX.D) -
                  FSPC (10,5)                         -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$ROOTD.$ROOT2.0DJ
 DELETE ($INDUV..$ROOTD.$ROOT2.0DJ) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$ROOTD.$ROOT2.0DJ) -
                  SHR (2,3) REUSE  NUMBERED           -
                  VOL ($VOLUV)  CYL (1,1)           -
                  RECSZ (180,180) )                 -
        DATA    ( NAME ($INDUV..$ROOTD.$ROOT2.0DJ.D) -
               CISZ (4096) )     /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$ROOTD.00DE
 DELETE ($INDSV..$ROOTD.00DE) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..$ROOTD.00DE) -
                  SHR (2,3) REUSE  KEYS (17,0)        -
                  VOL ($VOLSV)  CYL (7,1)           -
                  RECSZ (90,90) )                   -
        INDEX   ( NAME ($INDSV..$ROOTD.00DE.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDSV..$ROOTD.00DE.D) -
               CISZ (4096) )     /*: CATALOG ($SCAT) */
:/     ADD NAME=DF$ROOTD.$ROOT2.0DF
 DELETE ($INDSV..$ROOTD.$ROOT2.0DF) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..$ROOTD.$ROOT2.0DF) -
                  SHR (2,3) REUSE  KEYS (37,2)        -
                  VOL ($VOLUV)  CYL (5,1)             -
                  RECSZ (100,554) )                   -
        INDEX   ( NAME ($INDSV..$ROOTD.$ROOT2.0DF.I) -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDSV..$ROOTD.$ROOT2.0DF.D) -
                  FSPC (10,5)                         -
               CISZ (4096) )     /*: CATALOG ($SCAT) */
:/        ADD NAME=VERIFDA
 VERIFY FILE (PACDDA)
:/        ADD NAME=VERIFDC
 VERIFY FILE (PACDDC)
:/        ADD NAME=VERIFDX
 VERIFY FILE (PACDDX)
:/        ADD NAME=VERIFDJ
 VERIFY FILE (PACDDJ)
:/        ADD NAME=VERIFDE
 VERIFY FILE (PACDDE)
:/        ADD NAME=LI$ROOTD.$ROOT2.0DJ
 LISTCAT ENTRIES ($INDUV..$ROOTD.$ROOT2.0DJ) CLUSTER
:/     ADD NAME=MAXKEY
999999999999
:/     ADD NAME=REPRO999
 REPRO INFILE (MAXKEY)   OUTFILE (SYSPAF)
/*
//
```

19.4.3. GDG ALLOCATION/INITIALIZATION

3. GDG ALLOCATION/INITIALIZATION OF BACKUP FILES

JOB $prfj.2PR

CAUTION: If SMS is installed on your site, delete DD //GDGMOD from GDGBJ and GDGBB steps before executing the job.

DELETE : IDCAMS : DSCB MODEL file deletion,

BR14 : IEFBR14 : allocation of the DSCB MODEL,

GDGBJ : IDCAMS : GDG allocation of BJ Journal file,

INIBJ : IEBGENER : initialization of BJ Journal file,

GDGBB : IDCAMS : GDG allocation of BB backup file,

INIBB : IDCAMS : Copy of backed-up test deck on disk.

See the description in Subchapter 'USER FILES', Chapter: DSMS COMPONENTS.

```
//$PRFJ.2PR JOB ($CCPT),'PREPAR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*   JOB TO RUN ONLY THE FIRST TIME VA PAC DSMS IS INSTALLED          *
//*    . BUILDING OF DSCB MODEL AND INDEX DATA-GROUP FOR               *
//*      "BB" AND "BJ" SAVE FILES                                      *
//*    . "BJ" FILE INITIALIZATION                                      *
//*    . LOADING OF DSMS TEST FILES ON "BB" FILE                       *
//**********************************************************************
//*
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($DSCB)
//*
//BR14     EXEC PGM=IEFBR14
//DSCB     DD DISP=(,CATLG),UNIT=$UNITSN,VOL=SER=$VOLSN,SPACE=(TRK,0),
//            DSN=$DSCB
//GDGBJ    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD  DSN=$INDUN..$ROOTD.$ROOT2.0BJ,
//        DISP=(,KEEP,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,0),
//        DCB=($DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD *
  DEFINE GENERATIONDATAGROUP -
           (NAME ($INDUN..$ROOTD.$ROOT2.0BJ) LIMIT (3) SCR)
//*
//INIBJ     EXEC PGM=IEBGENER
//SYSIN    DD  DUMMY
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DUMMY,DCB=(RECFM=FB,LRECL=180,BLKSIZE=180)
//SYSUT2   DD  DSN=$INDUN..$ROOTD.$ROOT2.0BJ(+1),
//        DISP=(,CATLG,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,1),
//        DCB=($DSCB,RECFM=FB,LRECL=180,BLKSIZE=6300)
//*
//GDGBB    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD  DSN=$INDUN..$ROOTD.$ROOT2.0BB,
//        DISP=(,KEEP,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,0),
//        DCB=($DSCB,RECFM=VB,LRECL=354,BLKSIZE=6376)
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD *
  DEFINE GENERATIONDATAGROUP -
           (NAME ($INDUN..$ROOTD.$ROOT2.0BB) LIMIT (3) SCR)
//INIBB   EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//BBO  DD  DSN=$INDUN..$ROOTD.$ROOT2.0BB(+1),DISP=(,CATLG,DELETE),
//        UNIT=$UNITUN,VOL=SER=$VOLUN,SPACE=(TRK,60,RLSE),
//        DCB=($DSCB,RECFM=VB,LRECL=354,BLKSIZE=6376)
//BBI      DD  DSN=PAC.BB,DISP=OLD,UNIT=$U3480,
//             VOL=(,RETAIN,SER=$ICART),LABEL=(7,SL),
//             DCB=(RECFM=VB,LRECL=354,BLKSIZE=6376)
//SYSIN    DD  *
  REPRO INFILE (BBI)   OUTFILE (BBO)
//
```

19.4.4. LOADING OF LOAD-MODULES

```
4. ALLOCATION/LOADING OF THE LOAD-MODULE LIBRARIES

JOB $prfj.3PG
-------------

NOTE: If the library BLKSIZE is not suitable, modify its
size in the allocation STEP2 and replace the C(opy) by
COPYMOD in the load STEP3.

STEP1: IDCAMS   : library deletion,

STEP2: IEFBR14  : library allocation,

STEP3: IEBCOPY  : unloading of batch and on-line load-
                  modules.

See the description in Subchapter "ON-LINE/BATCH PROGRAM
LIBRARY", Chapter "DSMS COMPONENTS".

        CAUTION:   DO NOT FORGET TO CHOOSE THE xxUCTR
        =======    PROGRAM WHICH TRANSFORMS LOWERCASE
                   CHARACTERS INTO UPPERCASE CHARACTERS.
```

The installation of this program is described in the comments. Before running the Job, activate the corresponding line at the end of the STEP3 SYSIN:

CICS ESA 3.1 ==> S M=((PWUCTR,xxUCTR))

CICS ESA 3.2 ==> S M=((PWUCTX,xxUCTR))

CICS ESA 3.3 ==> S M=((PWUCTZ,xxUCTR))

If the PWUCTR program of CICS Rel. 3.1 is selected, it must be declared in assembler language in the CICS' CSD.

```
//$PRFJ.3PG JOB ($CCPT),'MBR8 MTR8 PDS',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*    INITIAL ALLOCATING OF THE MBR8 AND MTR8 PDS                    *
//*    OF THE BATCH AND ON-LINE PROGRAMS                              *
//*     .STEP1 : DELETE                                               *
//*     .STEP2 : ALLOCATION                                           *
//*     .STEP2 : LOADING                                              *
//*  ********************************************************         *
//*  WARNING : LOWER CASE UPPER CASE TRANSLATION INSTALLATION         *
//*  =======   ACTIVATE THE CORRESPONDING SELECT LINE:                *
//*  ********************************************************         *
//*   CICS ESA 3.1  ==>  S M=((PWUCTR,$ROOTD.UCTR))  ASSEMBLEUR       *
//*   CICS ESA 3.2  ==>  S M=((PWUCTX,$ROOTD.UCTR))  COBOL            *
//*   CICS ESA 3.3  ==>  S M=((PWUCTZ,$ROOTD.UCTR))  COBOL            *
//*                                                                   *
//*********************************************************************
//*
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($MODT)
 DELETE ($MODB)
//*
//STEP2    EXEC PGM=IEFBR14
//LNKB     DD DSN=$MODB,DISP=(,CATLG,DELETE),UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6144,(500,15,20)),DCB=(RECFM=U,BLKSIZE=6144)
//LNKT     DD DSN=$MODT,DISP=(,CATLG,DELETE),UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6144,(900,15,30)),DCB=(RECFM=U,BLKSIZE=6144)
//*
//STEP3   EXEC PGM=IEBCOPY
//************** LOADING BATCH AND TP PROGRAMS
//SYSPRINT DD  SYSOUT=$OUT
//SYSUT3   DD  UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTB     DD  DSN=$MODB,DISP=OLD        BATCH
//OUTP     DD  DSN=$MODT,DISP=OLD        ON LINE
//INB      DD  DSN=PACD.MBR8,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(5,SL)
//INT      DD  DSN=PACD.MTR8,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(6,SL)
//SYSIN    DD  *
  C I=((INB,R)),O=OUTB
===SEQ FOR DAF
  S M=DAFD10
  S M=PDSBDF
  S M=PDSDAC
===SEQ
  S M=PDCHOI
  S M=PDSA10
  S M=PDSB
  S M=PDSBAS
  S M=PDSBE
  S M=PDSCAM
  S M=PDSCEV
  S M=PDSCMQ
  S M=PDSCRQ
  S M=PDSCSI
  S M=PDSEX
  S M=PDSEXE
  S M=PDSE90
  S M=PDSFAC
  S M=PDSINI
  S M=PDSJMS
  S M=PDSLVB
  S M=PDSMS
  S M=PDSMSE
  S M=PDSRCT
  S M=PDSRFU
  S M=PDSRMS
  S M=PDSRQ0
  S M=PDSRQ1
  S M=PDSRQ2
  S M=PDSRQ3
  S M=PDSR10
  S M=PDSR20
  S M=PDSR30
  S M=PDSR40
===SEQ FOR ROLD
  S M=PDSR8B
  S M=PDSR8C
  S M=PDSR8Q
  S M=PDSR8R
  S M=PDSR8X
  S M=PDSR15
  S M=PDSR5J
===SEQ
  S M=PDSUAA
  S M=PDSUB1
  S M=PDSUB2
  S M=PDSUB3
  S M=PDSUB4
  S M=PDSUE1
  S M=PDSUE2
  S M=PDSUE3
  S M=PDSUK1
  S M=PDSUP0
  S M=PDSUP1
  S M=PDSUQ1
  S M=PDSUQ2
  S M=PDSUQ5
  S M=PDSUQ6
  S M=PDSUQ7
  S M=PDSUQ8
  S M=PDSUSI
  S M=PDSUS1
  S M=PDSUS3
  S M=PDSUS4
  S M=PDSUS6
  S M=PDSUTT
  S M=PDSUTV
  S M=PDSUTW
  S M=PDSUTX
  S M=PDSUTY
  S M=PDSUTZ
  S M=PDSUT1
  S M=PDSUT2
  S M=PDSUT3
  S M=PDSUT4
  S M=PDSUT5
  S M=PDSUT6
  S M=PDSUT7
  S M=PDSUT8
  S M=PDSUT9
  S M=PDSXCT
  S M=PDSXDT
  S M=PDSXST
  S M=PDSXTH
  S M=PDSXTR
  S M=PDS300
  S M=PDS320
  S M=((PDS381,PDS380))
  S M=PDS400
  S M=PDS450
  S M=PDS500
  S M=PDS600
  S M=PDS610
  S M=PDS700
  S M=PDS900
  S M=PTU001
  S M=REP2PJ
===SEQ FOR SEC
  S M=PACSECB
```

```
===SEQ
  C I=((INT,R)),O=OUTP
  S M=((PWCHOI,$ROOTD.CHOI))
  S M=((PWCUAM,$ROOTD.CUAM))
  S M=((PWCUEV,$ROOTD.CUEV))
  S M=((PWCUMQ,$ROOTD.CUMQ))
  S M=((PWCURQ,$ROOTD.CURQ))
  S M=((PWCUSI,$ROOTD.CUSI))
  S M=((PWMONI,$ROOTD.MONI))
  S M=((PWMOSO,$ROOTD.MOSO))
===SEQ FOR DAF
  S M=((PWTPDF,$ROOTD.TPDF))
===SEQ FOR SEC
  S M=((PWSECT,$ROOTD.SECT))
===SEQ
  S M=((PW00AA,$ROOTD.00AA))
  S M=((PW00AB,$ROOTD.00AB))
  S M=((PW00BA,$ROOTD.00BA))
  S M=((PW00B1,$ROOTD.00B1))
  S M=((PW00B2,$ROOTD.00B2))
  S M=((PW00B3,$ROOTD.00B3))
  S M=((PW00B4,$ROOTD.00B4))
  S M=((PW00B5,$ROOTD.00B5))
  S M=((PW00EA,$ROOTD.00EA))
  S M=((PW00E1,$ROOTD.00E1))
  S M=((PW00E2,$ROOTD.00E2))
  S M=((PW00E3,$ROOTD.00E3))
  S M=((PW00E4,$ROOTD.00E4))
  S M=((PW00E5,$ROOTD.00E5))
  S M=((PW00E6,$ROOTD.00E6))
  S M=((PW00FA,$ROOTD.00FA))
  S M=((PW00FB,$ROOTD.00FB))
  S M=((PW00HE,$ROOTD.00HE))
  S M=((PW00JO,$ROOTD.00JO))
  S M=((PW00KA,$ROOTD.00KA))
  S M=((PW00K1,$ROOTD.00K1))
  S M=((PW00K2,$ROOTD.00K2))
  S M=((PW00K3,$ROOTD.00K3))
  S M=((PW00LE,$ROOTD.00LE))
  S M=((PW00LS,$ROOTD.00LS))
  S M=((PW00MA,$ROOTD.00MA))
  S M=((PW00PA,$ROOTD.00PA))
  S M=((PW00P1,$ROOTD.00P1))
  S M=((PW00QA,$ROOTD.00QA))
  S M=((PW00QB,$ROOTD.00QB))
  S M=((PW00QC,$ROOTD.00QC))
  S M=((PW00Q1,$ROOTD.00Q1))
  S M=((PW00Q2,$ROOTD.00Q2))
  S M=((PW00Q3,$ROOTD.00Q3))
  S M=((PW00Q4,$ROOTD.00Q4))
  S M=((PW00Q5,$ROOTD.00Q5))
  S M=((PW00Q6,$ROOTD.00Q6))
  S M=((PW00Q7,$ROOTD.00Q7))
  S M=((PW00Q8,$ROOTD.00Q8))
  S M=((PW00Q9,$ROOTD.00Q9))
  S M=((PW00SA,$ROOTD.00SA))
  S M=((PW00SI,$ROOTD.00SI))
  S M=((PW00S1,$ROOTD.00S1))
  S M=((PW00S3,$ROOTD.00S3))
  S M=((PW00S4,$ROOTD.00S4))
  S M=((PW00S5,$ROOTD.00S5))
  S M=((PW00S6,$ROOTD.00S6))
  S M=((PW00S7,$ROOTD.00S7))
  S M=((PW00S8,$ROOTD.00S8))
  S M=((PW00S9,$ROOTD.00S9))
  S M=((PW00TA,$ROOTD.00TA))
  S M=((PW00TT,$ROOTD.00TT))
  S M=((PW00TU,$ROOTD.00TU))
  S M=((PW00TV,$ROOTD.00TV))
  S M=((PW00TW,$ROOTD.00TW))
  S M=((PW00TX,$ROOTD.00TX))
  S M=((PW00TY,$ROOTD.00TY))
  S M=((PW00TZ,$ROOTD.00TZ))
```

```
      S M=((PW00T1,$ROOTD.00T1))
      S M=((PW00T2,$ROOTD.00T2))
      S M=((PW00T3,$ROOTD.00T3))
      S M=((PW00T4,$ROOTD.00T4))
      S M=((PW00T5,$ROOTD.00T5))
      S M=((PW00T6,$ROOTD.00T6))
      S M=((PW00T7,$ROOTD.00T7))
      S M=((PW00T8,$ROOTD.00T8))
      S M=((PW00T9,$ROOTD.00T9))
      S M=((PW00UD,$ROOTD.00UD))
      S M=((PWR980,$ROOTD.R980))
    //*  S M=((PWUCTR,$ROOTD.UCTR))
    //*  S M=((PWUCTX,$ROOTD.UCTR))
    //*  S M=((PWUCTZ,$ROOTD.UCTR))
    //
```

19.4.5. LOADING OF ERROR MESSAGES

5. DEFINITION/LOADING OF ERROR MESSAGES FILE/HELP FUNCTION AND PAF WORK FILE

JOB $prfj.4DE

DELDEF : IDCAMS : DELETE/DEFINE of the DE and DF files.

REPRO : IDCAMS : REPRO of the DE file.

See the description in Subchapter 'SYSTEM FILES', Chapter: 'DSMS COMPONENTS.

```
//$PRFJ.4DE  JOB ($CCPT),'DSMS DE FILE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*    LOADING OF VA PAC-DSMS ERROR MESSAGE FILE
//*    AND PAF WORK FILE
//*********************************************************************
//DELDEF   EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//*****  DELETE DEFINE 'DE' FILE
//*
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD DSN=$INDSN..$ROOTD.$ROOTD.SY(DF$ROOTD.00DE),DISP=SHR
//      DD DSN=$INDSN..$ROOTD.$ROOTD.SY(DF$ROOTD.$ROOT2.0DF),
//      DISP=SHR
//REPRO    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//*****  LOADING 'DE' FILE
//*
//SYSPRINT DD SYSOUT=$OUT
//DEO      DD  DSN=$INDSV..$ROOTD.00DE,DISP=SHR
//DEI      DD  DSN=PAC.DE,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(4,SL),
//         DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSIN    DD  *
  REPRO INFILE (DEI)   OUTFILE (DEO)
//*
//REPRO    EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//*
//SYSPRINT DD SYSOUT=$OUT
//DFO      DD  DSN=$INDSV..$ROOTD.$ROOT2.0DF,DISP=SHR
//DFI      DD  *
9999999999999999999999999999999999999999
/*
//SYSIN    DD  *
  REPRO INFILE (DFI)   OUTFILE (DFO)
//*
```

19.4.6. LOADING OF BATCH PROCEDURES

6. CATALOGING OF OPERATIONS PROCEDURES: IEBUPDTE

It is recommended that all DSMS operation procedures be cataloged in a PROCLIB. The '$prfj.5PR' job called 'PROCEDURES' creates one member per procedure.

```
CAUTION: change :/ characters with ./ before
         executing the job.
```

Each member is coded '$radp.XXXX', where $radp is the root selected at installation and XXXX is the standard IBM name for the procedure. The procedures are described in Chapter: BATCH PROCEDURES.

```
//$PRFJ.5PR JOB ($CCPT),'PROCEDURES ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*       ---- CATALOGING PROCEDURES ----                             *
//*            =====================                                  *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*********************************************************************
//  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2 DD DSN=$BIBP,DISP=SHR
//SYSIN   DD   DATA,DLM='F+'
:/ ADD NAME=$RADP.DARC
:/ ADD NAME=$RADP.DEXH
:/ ADD NAME=$RADP.DEXP
:/ ADD NAME=$RADP.DEXQ
:/ ADD NAME=$RADP.DEXT
:/ ADD NAME=$RADP.DINI
:/ ADD NAME=$RADP.DINS
:/ ADD NAME=$RADP.DLVB
:/ ADD NAME=$RADP.DPDF
:/ ADD NAME=$RADP.DPRT
:/ ADD NAME=$RADP.DREN
:/ ADD NAME=$RADP.DREO
:/ ADD NAME=$RADP.DRST
===SEQ FOR ROLD
:/ ADD NAME=$RADP.DR80
:/ ADD NAME=$RADP.DR8X
:/ ADD NAME=$RADP.DR8Q
:/ ADD NAME=$RADP.DR15
:/ ADD NAME=$RADP.DR5J
===SEQ
:/ ADD NAME=$RADP.DSAV
:/ ADD NAME=$RADP.DUPT
:/ ADD NAME=$RADP.DXBJ
===SEQ
F+
//
```

19.4.7. LOADING OF DSMS TEST DATABASE

```
7. LOADING OF THE TEST DECK FILES

JOB $prfj.6DS
-------------

DRST   : $radp.DRST : restoration of the test deck files.
```

See the description in Subchapter 'USER FILES', Chapter: DSMS COMPONENTS.

```
//$PRFJ.6DS   JOB ($CCPT),'TEST FILES',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*             - LOADING TEST FILES -                                *
//*********************************************************************
//DRST    EXEC $RADP.DRST
 RE
//
```

19.4.8. CICS CHANGES

8. MODIFICATION OF JCL FOR SUBMISSION OF CICS

The CICS submission JCL might have to be completed in order to take all DSMS elements into account:

```
//DFHRPL   On-line load module-library (MTR8)
```

TRANSACTION OUTPUT MODIFICATION

The system sends a CICS RETURN as output of the xxyE DSMS transaction, by the xxMOSO program called by XCTL.

If it seems more convenient, you may replace this program with a program of your own, for instance to return control to a user general menu or any other application.

See Subchapter 'CICS Environment', in Chapter: ENVIRONEMENT.

19.4.9. COMPLEMENT - USER ENTRY-POINTS SOURCE FILE

9. INSTALLATION OF SOURCES FOR USER CHECKS

This complementary installation should only be performed by users who wish to add checks on the change, event, query, layout or site definition screens.

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.)

```
//$PRFJ.7SR  JOB ($CCPT),'SOURCES',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*******************************************************************
//* VisualAge Pacbase-DSMS                                          *
//*                                                                 *
//*             - INSTALLATION - DI7SRC -                           *
//*                                                                 *
//*     .STEP1 : DELETE                                             *
//*     .STEP2 : ALLOCATION                                         *
//*     .STEP3 : LOADING                                            *
//*  ->NOTE                                                         *
//*    ----                                                         *
//*   THIS PDS CONTAINS USER CONTROL SUB-PROGRAMS                   *
//*******************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..$ROOTD.$ROOTD.SRC)
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDSN..$ROOTD.$ROOTD.SRC,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(6080,(100,10,10))
//*
//STEP3  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OUTM     DD DSN=$INDSN..$ROOTD.$ROOTD.SRC,DISP=OLD
//INM      DD DSN=PACD.SOURCE,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(8,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,$ROOTD.CUAM))
  S M=((PWCUEV,$ROOTD.CUEV))
  S M=((PWCUMQ,$ROOTD.CUMQ))
  S M=((PWCURQ,$ROOTD.CURQ))
  S M=((PWCUSI,$ROOTD.CUSI))
  S M=PACTSS
  S M=PACTSSC
  S M=((DAFDICA,DAFDIC))
//
```

9.2. INSTALLATION OF THE DAF ENVIRONMENT

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

The PAF Preprocessor processes the generated programs in order to perform this transformation. 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.)

Three DAF sub-programs are provided in the installation deck:

. Two batch sub-programs installed in the batch load-module library:

- PDSBDF for DAF standard requests
- PDSDAC for physical access to the DSMS database

> NOTE:
> These sub-programs 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).

. One on-line sub-program, xxTPDF, installed in the on-line module library.

The work file necessary for the operation of DAF in on-line mode has a CICS-imposed DDNAME of the $rootd.$root2.0DF format. This DDNAME must be unique 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.

```
//*******************************************************************
//* VisualAge Pacbase-DSMS                                          *
//*                                                                 *
//*               - JCL EXAMPLE -                                   *
//*    EXECUTION OF A USER D.A.F. BATCH PROGRAM                     *
//*******************************************************************
//DAFBATCH PROC ROOTD=$ROOTD, 2 FIRST CHARACTERS TRANSACTION
//      ROOT2=$ROOT2,         3RD CHARACTER TRANSACTION CODE
//      INDSV='$INDSV',       INDEX OF SYSTEM VSAM FILES
//      INDUV='$INDUV',       INDEX OF USER VSAM FILES
//      INDSN='$INDSN',       INDEX OF SYSTEM NON VSAM FILES
//*:    VSAMCAT='$VCAT',      USER VSAM CATALOG
//*:    SYSTCAT='$SCAT',      SYSTEM VSAM CATALOG
//      STEPLIB=,             USER LIBRARY OF LOAD-MODULES
//      OUT=$OUT              OUTPUT CLASS
//*******************************************************************
//MAXKEY  EXEC PGM=IDCAMS
//***********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&&SYSDAF,DISP=(NEW,KEEP),
//         SPACE=(CYL,(3,3)),
//         LRECL=420,RECORG=KS,KEYOFF=0,KEYLEN=12
//MAXKEY   DD DSN=&INDSN..&ROOTD.&ROOTD.SY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..&ROOTD.&ROOTD.SY(REPRO999),DISP=SHR
//WITHDAF  EXEC PGM=------
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//PACDDA   DD DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDC   DD DSN=&INDUV..&ROOTD.&ROOT2.0DC,DISP=SHR
//PACDDX   DD DSN=&INDUV..&ROOTD.&ROOT2.0DX,DISP=SHR
//PACDDE   DD DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//SYSDAF   DD DSN=&&SYSDAF,DISP=(OLD,PASS)
//------   DD DSN=---
//------   DD DSN=---
//------   DD DSN=---
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//         PEND
//DAFBATCH EXEC DAFBATCH
```

19.4.10. COMPLEMENT - SECURITY SYSTEMS INTERFACE

```
10. SECURITY SYSTEM INTEFACE

Job '$prfj.8SR'

STEP2: IEBCOPY  : TP sub-routine loading


Job '$prfj.8SER' for RACF

STEP1: IEBCOPY  : loading of the PACSECU8 sub-routine
                  which accesses the RACF Tables in an
                  AUTHORIZED LIBRARY.
```

INSTALLATION COMPLEMENTS

(Details in the SECURITY SYSTEM INTERFACE Manual).

RACF

It is not necessary to define resources classes because checks will only be performed on user code and password. TOPSECRET

Access sub-routines compilation.

PACTSS & PACTSSC must be compiled with the 'OPMAT' TSS library in the SYSLIB carte of the Assembler compilation.

PACTSSC is a CICS program and must be transfered before compilation and link-edit.

PACTSSC and the TSSCAI program (Computer Associates) must be declared in the CICS CSD and located in one of the DFHRPL load-module library.

The sources are stored in the source PDS loaded in the previous step.

CAUTION:

For the checks associated to the security systems to be performed in DSMS, the DRST procedure must be run with the corresponding options.

```
===SEQ FOR SEC
//$PRFJ.8SR JOB ($CCPT),'DSMS DI8SER',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*******************************************************************
//* VisualAge Pacbase-DSMS                                          *
//*                                                                 *
//*             - INSTALLATION - DI8SER RACF -                      *
//*                                                                 *
//*    LOADING OF THE RACF SECURITY SYSTEM INTERFACE MODULE         *
//*     .STEP1 : LOADING                                            *
//*******************************************************************
//STEP1    EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(CYL,(2,1))
//OULIB    DD DSN=-----,DISP=OLD         <-- AUTHORIZED LIBRARY
//INLIB    DD DSN=PACD.MBR8,DISP=SHR,
//         VOL=(,RETAIN,SER=$ICART),UNIT=$U3480,LABEL=(5,SL)
//SYSIN    DD *
  C I=((INLIB,R)),O=OULIB
  S M=((PACSECRA,PACSECU8))
/*
//
===SEQ
```

19.5. LIST OF INSTALLED PROGRAMS

LIST OF PROGRAMS AND INSTALLED FILES

===MOD I99INSL

This list, result of the '$prfj.DINS' job executing the INSL procedure, contains:

. The list of the batch and on-line programs with their compilation dates,

. The list of the permanent system files DA and DS with their date of creation.

Keep this list in order to send IBM the installation references in case of System problems.

User Input:

```
-------------------------------------------------------
!Col.! Len.! Value  ! Designation                     !
!----+-----+--------+---------------------------------!
!  3 !   2 !  rr    ! rr=ROOT DSMS module root        !
-------------------------------------------------------
```

Return codes:

4 INSUFFICIENT MEMORY (increase the REGION parameter)
6 NO LOAD MODULE IN THE LIBRARY
A NOT STANDARD

Contact IBM if one one the following errors occurs:

5 UNKNOWN ERROR
7 INPUT/OUTPUT ERROR
8 SYSTEM ERROR
8 LOADING ERROR (SYSTEM ERROR)

```
//$PRFJ.DIN  JOB ($CCPT),'PROGRAMS DATE',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//***********************************************************
//*         LIST OF PROGRAM DATES                           *
//***********************************************************
//DINS   EXEC $RADP.DINS
  $ROOTD
```

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - LIST OF INSTALLED MODULES -                        *
//*********************************************************************
//$RADP.DINS  PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',   3RD CHARACTER TRANSACTION CODE
//  INDUV='$INDUV',   INDEX OF USER VSAM FILES
//  INDSV='$INDSV',   INDEX OF SYSTEM VSAM FILES
//  STEPLIB='$MODB',  LIBRARY BATCH LOAD MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//  MODT='$MODT',     LIBRARY TP LOAD MODULES
//  OUT='$OUT',       OUTPUT CLASS
//  UWK='$UWK'        UNIT OF WORK FILES
//*********************************************************************
//*   INPUT:
//*   - 1ST LINE: 2 FIRST CHARACTERS TRANSACTION CODE (COL.3)
//*   - 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=&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=3200)
//PDSXDT EXEC  PGM=PDSXDT
//***********************
//STEPLIB  DD  DSN=&STEPLIB,DISP=SHR
//         DD  DSN=&MODT,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD  SYSOUT=&OUT
//PACDDA   DD  DSN=&INDUV..&ROOTD.&ROOT2.0DA,DISP=SHR
//PACDDE   DD  DSN=&INDSV..&ROOTD.00DE,DISP=SHR
//PACDDS   DD  SYSOUT=&OUT
//PACDMB   DD  DSN=&&PAC7MB,DISP=(OLD,PASS)
//SYSUDUMP DD  SYSOUT=&OUT
```

19.6. SETTING OF A VA PAC DATABASE UNDER DSMS CONTROL

SETTING OF A VA PAC DATABASE UNDER DSMS CONTROL

To place a VA Pac Database under DSMS control, access the PB screen of the xxEE transaction (VA Pac user parameters update transaction).

Associate the DSMS transaction code ('DSMS base code') with the VA Pac logical Database code. It is possible to associate one DSMS Database code to several VA Pac codes, or to associate one DSMS Database code to one VA Pac code.

The minimum authorization level required to perform this update is 3.

ADDITIONAL USER OPERATIONS

The DSMS module includes a functionality whereby VA Pac must consult the DSMS VA Pac Elements file (DC).

If the DSMS Function is in the same CICS as the VA Pac Database using it, no additional user operation is necessary.
If the DSMS Function and the VA Pac Database are not in the same CICS, the DC file must be declared under the CICS in which the VA Pac Database resides, as follows:

. in Read Only mode (READ, BROWSE).

If the DSMS module and the VA Pac Database are located at different sites, a copy of the DC file must be installed at VA Pac site (via an IDCAMS, DELETE/DEFINE, and REPRO). This copy will have to be "refreshed" regularly in function with the updates made on the DC file.

These operations are the user's responsibility, as no JCL is provided.

19.7. UTILIZATION TESTS

UTILIZATION TESTS

DSMS INSTALLATION TESTS

These tests have three steps:

. On-line utilization tests,

. Extraction utility test,

. Database management tests.

1. ON-LINE UTILIZATION TESTS

Open the test Database files in on-line mode. Logon with the user TEST. Use password 'CGI'. 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 utilization tests journal: 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 quickly On-Line tests.

19.7.1. JCL TEST : 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
//
```

19.7.2. JCL TEST : DARC

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

19.7.3. JCL TEST : DSAV

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

19.7.4. JCL TEST : 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
//*
```

19.7.5. JCL TEST : 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. STANDARD REINSTALLATION

SYSTEM REINSTALLATION

When you receive a sub-release with abends corrections or punctual developments, DSMS must be installed again.

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

. the product installation cardridge (or tape),

. the list of the corrected abends,

. additional instructions might be included for the reinstallation procedure, when the procedure described in this Subchapter is not enough.

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:

- DI3PGM: $prfj.3PG programs initialization.

- DI4DE: $prfj.4DE error messages initialization.

1) DI3PGM: 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 comming 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) DI4DE: 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 DI3PGM

===SELM DI4DE
```

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.

20. RETRIEVAL OF DSMS 8.0.1 & ADAPTATION TO 2.5 (DR80)

20.1. INTRODUCTION

INTRODUCTION

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 back-up 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

```
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


CONVERSION TO 2.5:  PDSR8C

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

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

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

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


.Sort files:
 SORTWK01, SORTWK02, SORTWK03

.Output report:
 -Retrieval report
  PACDIO

CONVERSION OF BACKUP TO 2.5 FORMAT:  PDSR15

.Permanent input file:
 -Backup file from preceding step
  PACDBB  :    DSN=&BBOLD

.Permanent output file:
 -Backup of converted DSMS Database
  PACDB1  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1)

.Output report:
 -Printing report
  PACDIK
```

20.4. PROCEDURE JCL

```
//*******************************************************************
//* VisualAge Pacbase-DSMS                                          *
//*                                                                 *
//*              - 8.0.1 DSMS DATABASE RETRIEVAL -                  *
//*******************************************************************
//$RADP.DR80 PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  SORTLIB='$BIBT',       SORT LIBRARY
//  CYL=3,                 SORTWORK SPACE
//  UWK=$UWK,              WORK UNIT
//  OLDBB=,                8.0.1 DSMS BACKUP DSNAME
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  SPAWK='(CYL,(20,2))',  WORK FILE SPACE
//  SPABB='(TRK,(45,5),RLSE)'   SPACE OF BACKUP (IF DISK)
//*******************************************************************
//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=3200)
//PDSR8B EXEC PGM=PDSR8B,REGION=2048K
//**********************
//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
//PACDIK   DD SYSOUT=&OUT
//PACDIQ   DD DSN=&&PACDIQ,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAWK,
//         DCB=(RECFM=FB,LRECL=284,BLKSIZE=5680)
//PACDIT   DD DSN=&&PACDIT,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAWK,
//         DCB=(RECFM=FB,LRECL=104,BLKSIZE=2080)
//PACDIW   DD DSN=&&PACDIW,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAWK,
//         DCB=(RECFM=FB,LRECL=92,BLKSIZE=2024)
//PACDLA   DD DSN=&&PACDLA,DISP=(,PASS),
//         UNIT=&UWK,SPACE=&SPAWK,
//         DCB=(RECFM=FB,LRECL=1,BLKSIZE=1000)
//PACDMB   DD DSN=&&DR80MB,DISP=(OLD,PASS)
//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
//PDSR8C EXEC PGM=PDSR8C,REGION=2048K
//**********************
//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
//PACDB1    DD DSN=&&BBOLD,DISP=(,PASS),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDIO   DD SYSOUT=&OUT
//PACDIQ   DD DSN=&&PACDIQ,DISP=(OLD,PASS)
//PACDIT   DD DSN=&&PACDIT,DISP=(OLD,PASS)
//PACDIW   DD DSN=&&PACDIW,DISP=(OLD,PASS)
//PACDLA   DD DSN=&&PACDLA,DISP=(OLD,PASS)
//PACDMB   DD DSN=&&DR80MB,DISP=(OLD,PASS)
```

```
//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
//PDSR15 EXEC PGM=PDSR15,REGION=2048K
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDBB   DD DSN=&&BBOLD,DISP=(OLD,PASS)
//PACDB1    DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDIK   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

21. RETRIEVAL DSMS 8.0.2 01/02 & ADAPT. TO 2.5 (DR8Q)

21.1. OPERATIONS TO CARRY OUT

INTRODUCTION

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.

Installation of the new DSMS release calls for the retrieval of 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)

Using the new procedures:

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

EXECUTION CONDITIONS

None.
However, for coherence purposes it is advisable to close the retrieved 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 8.0.2 01/02 DSMS Database
  PACDBB  :    DSN=&OLDBB

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

CONVERSION OF BACKUP TO 2.5 FORMAT:  PDSR15

.Permanent input file:
 -Backup file from preceding step
  PACDBB  :    DSN=&BBOLD

.Permanent output file:
 -Backup of converted DSMS Database
  PACDB1  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1)

.Output report:
 -Printing report
  PACDIK
```

21.3. PROCEDURE JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*        - 8.0.2 01 / 02 DSMS DATABASE RETRIEVAL INTO 2.5 -         *
//*********************************************************************
//$RADP.DR8Q PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  SORTLIB='$BIBT',       SORT LIBRARY
//  CYL=3,                 SORTWORK SPACE
//  UWK=$UWK,              WORK UNIT
//  OLDBB=,                8.0.2 02 DSMS BACKUP DSNAME
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       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,REGION=2048K
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDBB   DD DSN=&&BBOLD,DISP=(OLD,PASS)
//PACDB1    DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDIK   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

22. RETRIEVAL OF 8.0.2, PACBASE 8.0.1-COMPATIBLE DSMS

22.1. PRESENTATION

INTRODUCTION

The current release of DSMS is not compatible with PACBASE release 8.0.1.

If DSMS 8.0.2 was being used with PACBASE 8.0.1, the installation of the new version of DSMS requires the following operations to be carried out:

1. DSMS Database backup (DSAV).

2. Installation of DSMS version 1.5.

3. Installation of PACBASE version 8.0.2, 1.2 or 1.5.

4. Retrieval of the BB backup file (DR8X procedure).

5. Reorganization of the DSMS Database to rebuild the DX file.

6. DSMS Database restoration.

EXECUTION CONDITIONS

None. However, to ensure the integrity of the retrieved Database, it is advised to close the Database to On-Line process.

22.2. PROCEDURE 'DR8X' - USER INPUT

USER INPUT

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

POS.	LEN.	VALUE	MEANING
1	3	'PRO'	Old product code
4	3	'PRO'	New product code Each group of 6 characters can be repeated a maximum of 13 times per line.

22.3. PROCEDURE 'DR8X'

```
CONVERSION FOR PACBASE 8.0.2:  PDSR8X

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

.Sort files:
 SORTWK01, SORTWK02, SORTWK03, SORTWK04

.Input transaction file:
  PACDMB  :    DSN=&&DR8XMB

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

.Output report
 -Retrieval report
  PACDIK

CONVERSION OF BACKUP TO 2.5 FORMAT:  PDSR15

.Permanent input file:
 -Backup file from preceding step
  PACDBB  :    DSN=&BBOLD

.Permanent output file:
 -Backup of converted DSMS Database
  PACDB1  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1)

.Output report:
 -Printing report
  PACDIK
```

22.4. PROCEDURE 'DR8X' - JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//* - RETRIEVAL OF 8.0.2 DSMS DATABASE CONNECTED TO PACBASE 801 -     *
//*                 INTO 2.5 DSMS DATABASE                            *
//*********************************************************************
//$RADP.DR8X PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  SORTLIB='$BIBT',       SORT LIBRARY
//  CYL=3,                 SORTWORK SPACE
//  UWK=$UWK,              WORK UNIT
//  OLDBB=,                8.0.1 DSMS BACKUP DSNAME
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       BACKUP UNIT (DISK OR TAPE)
//  SPABB='(TRK,(45,5),RLSE)'   SPACE OF BACKUP (IF DISK)
//*********************************************************************
//INPUT  EXEC PGM=PTU001
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&DR8XMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//PDSR8X EXEC PGM=PDSR8X,REGION=2048K
//**********************
//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
//PACDB1    DD DSN=&&BBOLD,DISP=(OLD,PASS),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDIK   DD SYSOUT=&OUT
//PACDMB   DD DSN=&&DR8XMB,DISP=(OLD,DELETE)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK04 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PDSR15 EXEC PGM=PDSR15,REGION=2048K
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDBB   DD DSN=&&BBOLD,DISP=(OLD,PASS)
//PACDB1    DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDIK   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

22.5. DATABASE RETRIEVAL JOB

```
//$PRFJ.82B  JOB ($CCPT),'DR8X',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//* --- RETRIEVAL OF 8.0.2 DSMS DATABASE CONNECTED TO PACBASE 8.0.1  *
//*     INTO 2.5 DSMS DATABASE                                       *
//********************************************************************
//DR8X   EXEC $RADP.DR8X,
//  OLDBB=     <- 8.0.1 PACBASE COMPATIBLE 8.0.2 DSMS BACKUP DSNAME
//
//$PRFJ.82C  JOB ($CCPT),'DREO',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*
//*********************************************************************
//*             - REORGANIZATION -                                    *
//*     WARNING : TO BE SUBMITTED ONLY IF PRODUCT CODES              *
//*     WERE CHANGED IN THE "DR8X" EXECUTION                          *
//*********************************************************************
//DREO   EXEC $RADP.DREO
//
```

23. RETRIEVAL OF DSMS 1.2 OR 1.5 AS DSMS 2.5

23.1. INTRODUCTION

INTRODUCTION

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 back-up in order to rebuild the DX cross-references in the new 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.

23.2. PROCEDURE '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=&INDUN..&ROOTD.&ROOT2.0BB(+1)

.Output report:
 -Printing report
  PACDIK
```

23.3. PROCEDURE 'DR15' - JCL

```
//**********************************************************************
//* VisualAge Pacbase-DSMS                                             *
//*                                                                    *
//*              - 1.2 / 1.5 DSMS DATABASE RETRIEVAL -                 *
//*                          INTO 2.5                                  *
//**********************************************************************
//$RADP.DR15 PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',         INDEX OF USER NON VSAM FILES
//  STEPLIB='$MODB',        LIBRARY OF LOAD MODULES
//  DSCB='$DSCB',           DSCB MODEL FILE
//  OUT='$OUT',             OUTPUT CLASS
//  OLDBB=,                1.2 OR 1.5 DSMS BACKUP DSNAME
//  VOLS='SER=$VOLUN',      VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',        BACKUP UNIT (DISK OR TAPE)
//  SPABB='(TRK,(45,5),RLSE)'     SPACE OF BACKUP (IF DISK)
//****************************************************************
//PDSR15 EXEC PGM=PDSR15,REGION=2048K
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDBB   DD DSN=&OLDBB,DISP=OLD
//PACDB1    DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//PACDIK   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

23.4. DATABASE RETRIEVAL JOB

```
===SEQ FOR ROLD
//$PRFJ.15A  JOB ($CCPT),'DR15',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - 1.2 OR 1.5 DSMS DATABASE RETRIEVAL -               *
//*********************************************************************
//DR15   EXEC $RADP.DR15,
//  OLDBB=     <- OLD DSMS BACKUP DSNAME
//
//$PRFJ.15B  JOB ($CCPT),'DREO',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*
//*********************************************************************
//*              - REORGANIZATION -                                   *
//*********************************************************************
//DREO   EXEC $RADP.DREO
//
```

23.5. PROCEDURE 'DR5J' - 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
  PACDJB  :    DSN=&INDUN..&ROOTD.&ROOT2.0BJ(+1)

.Output report:
 -Printing report
  PACDIK
```

23.6. PROCEDURE 'DR5J' - JCL

```
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*   - RETRIEVAL OF A 1.2 OR 1.5 DSMS JOURNAL -                      *
//*                                                                   *
//*********************************************************************
//$RADP.DR5J PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',         INDEX OF USER NON VSAM FILES
//  STEPLIB='$MODB',        LIBRARY OF LOAD MODULES
//  DSCB='$DSCB',           DSCB MODEL FILE
//  OUT='$OUT',             OUTPUT CLASS
//  OLDBJ=,                1.2 OR 1.5 DSMS SEQUENTIAL JOURNAL
//  VOLS='SER=$VOLUN',      VOLUME OF BACKUP (JB)
//  UNITS='$UNITUN',        BACKUP UNIT (DISK OR TAPE)
//  SPABJ='(TRK,(45,5),RLSE)'     SPACE OF BACKUP (IF DISK)
//***************************************************************
//PDSR5J EXEC PGM=PDSR5J,REGION=2048K
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//PACDBJ   DD DSN=&OLDBJ,DISP=OLD
//PACDJB    DD DSN=&INDUN..&ROOTD.&ROOT2.0BJ(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABJ,
//          DCB=(&DSCB,RECFM=FB,BLKSIZE=6300,LRECL=180)
//PACDIK   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

23.7. JOURNAL FILE RETRIEVAL JOB

```
===SEQ FOR ROLD
//$PRFJ.15A  JOB ($CCPT),'DR5J',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VisualAge Pacbase-DSMS                                            *
//*                                                                   *
//*              - 1.2 OR 1.5 DSMS ARCHIVED JOURNAL RETRIEVAL -       *
//*********************************************************************
//DR5J   EXEC $RADP.DR5J,
//  OLDBJ=     <- OLD ARCHIVED JOURNAL
//
```

24. REPLACEMENT OF LOW-VALUES WITH BLANKS (DLVB)

24.1. DLVB: REPLACEMENT OF LOW-VALUES WITH BLANKS

INTRODUCTION

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 it pssible to transfer the BB file onto different platforms, while avoiding problems due to the presence of low-values at the time of transfer.

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).

EXECUTION CONDITION

None

24.2. DLVB: PARAMETERS-DESCRIPTION OF STEPS

DLVB: DESCRIPTION OF STEPS

```
REPLACEMENT OF LOW-VALUES WITH 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=&INDUN..&ROOTD.&ROOT2.0BB(0)

.Output file:
 -New Database backup
  PACDB1  :    DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1)
```

24.3. DLVB: EXECUTION PROCEDURE

```
//*******************************************************************
//* VisualAge Pacbase-DSMS                                          *
//*                                                                 *
//* - REPLACEMENT OF LOW VALUE CHARACTERS WITH BLANK CHARACTERS -   *
//*                                                                 *
//*  OPTION : SUBMIT PROCEDURE WITH PARM='DATA' TO PROCESS DATA ONLY  *
//*******************************************************************
//$RADP.DLVB PROC ROOTD='$ROOTD', 2 FIRST CHARACTERS TRANSACTION CODE
//  ROOT2='$ROOT2',        3RD CHARACTER TRANSACTION CODE
//  INDUN='$INDUN',        INDEX OF USER NON VSAM FILES
//  STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//*:  VSAMCAT='$VCAT',    USER VSAM CATALOG
//*:  SYSTCAT='$SCAT',    PACBASE DSMS SYSTEM VSAM CATALOG
//  DSCB='$DSCB',          DSCB MODEL FILE
//  OUT='$OUT',            OUTPUT CLASS
//  VOLS='SER=$VOLUN',     VOLUME OF BACKUP (BB)
//  UNITS='$UNITUN',       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=&INDUN..&ROOTD.&ROOT2.0BB(0),DISP=SHR
//PACDB1   DD DSN=&INDUN..&ROOTD.&ROOT2.0BB(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPABB,
//            DCB=(&DSCB,RECFM=VB,BLKSIZE=6376,LRECL=354)
//SYSUDUMP DD SYSOUT=&OUT
```