VisualAge Pacbase 2.5

VA PAC 2.5 : IBM IMS/VS
OPERATIONS MANUAL VOLUME II : ADMINISTRATOR'S GUIDE

DEPIM002251A

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 (August 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. OVERVIEW

THE ADMINISTRATOR'S GUIDE: OVERVIEW

This manual contains the descriptions of all the Batch procedures used by a VisualAge Pacbase Database Administrator.

These procedures relate mainly to the following operations fields:

- Database management
- Versioning (PEI and Pac/Transfer)
- Manager's utilities
- Migrations

PRESENTATION OF PROCEDURES

Batch processing is divided into various procedures. The following chapters describe these procedures and their specific execution conditions.

The presentation of a procedure contains the following:

. General introduction, including

  -presentation

  -execution condition(s)

  -actions to be taken in case of abnormal execution

. Descriptions of user input, processing, results, and possible recommendations

. Execution JCL.

1.1. USER IDENTIFICATION (*)

USER IDENTIFICATION '*' LINE

Batch procedures which access the Database require a user identification ('*'-type) line at the beginning of user input to identify the user as well as the library and session in which he/she wishes to work. (There may be several '*'-type lines if the procedure applies to several libraries; see the description of each procedure's user input.)

Some information entered on this screen is the same as that entered on the Sign-On screen. It is thus possible to check if the user's commands are compatible with his/her authorizations.

Before running any batch procedure, the user must make sure he/she has the adequate authorization level. Authorization levels are defined by the Database administrator, using the PARM User Parameter Management procedure.

POS.	LEN.	VALUE	MEANING
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Library code
22	4	ssss	Session number
26	1		Version of the session:
		'H'	Frozen session
		'T'	Test session
27	1		With the UPDT procedure, in case of multiple deletion:
		'N'	Print all transations including implicit transactions (Default)
		'O'	Print entered transactions and erroneous transactions
		'E'	Print erroneous transactions only

POS.	LEN.	VALUE	MEANING
28	1		Language code (F or A)
29	11		DO NOT USE
			The two following fields are to be entered for all procedures generating update transactions which will modify a library or session under DSMS control.
			You may also enter them on the '*' line of UPDT.
40	3		PRODUCT CODE (on 3 characters)
43	6		CHANGE NUMBER (on 6 characters, the non-significant zeros must be entered).
			These two codes will be displayed in the Journal after the execution of UPDT.
49	1		TRANSFER OF OCCURRENCE LOCK:
		'Blank'	Replacement of the code of the user who locked the entity with that found on the '*' line.
		1	The new entities created from the extracted entities are not locked after UPDT
		2	The code of the user who locked the entities is kept
50	1		TRANSFER OF THE PASSWORD on the extraction prodedures, in the '*'-line at the top of the generated output transactions:
		'Blank'	Password is not transferred in the output file.
		1	Password is transferred.
			NOTE: For EXTR, the '*' line is transferred in the output file only if you input 'C' in position 1.

Some of the information entered on a '*' line is entered on the Sign-on screen. For more details, refer to Chapter 'USING THE SYSTEM ON-LINE', Subchapter 'Conversation Initialization/ Sign-on', in the VisualAge Pacbase Interface User's Guide.

1.2. ACCESS RIGHTS: BATCH-PROCEDURE AUTHOR. OPTION

'BATCH-PROCEDURE ACCESS AUTHORIZATION' OPTION

PRINCIPLE OF THE OPTION

This option is used to grant each user the access.

For example, a user needs authorization level 4 for database management procedures (such as MLIB or REST) and authorization level 2 for element-extraction procedures (such as PACX).

This authorization level assignment is performed using the PARM procedure. The level can take a value from 4 to 0.

When the option is active, the system allows you to grant each user:

- a global level of authorization for access to the batch procedures,
- a database level of authorization for access to the batch procedures (platforms allowing management of several user databases for one system).

CONSEQUENCE

The option requires a '*' line with user code and password as input of the procedures checked for access authorizations.

OPTION ACTIVATION

For VisualAge Pacbase installation, the option activation is not a default setting. It must be done through an update of the user parameters:

. in batch mode: 'NS' line of the PARM procedure;
. in on-line mode: 'PK' screen.

Authorization levels for all procedures are described in the following table, and mentioned in the "Execution Conditions" paragraph for each procedure.

BATCH PROCEDURE ACCESS AUTHORIZATION TABLE

PROCEDURE	GLOBAL AUTHORIZATION	DATABASE AUTHORIZATION
MLIB	4	
REST	4	
SAVE	4	
REOR	4	
ARCH	4	
REAG	4	
SVAG	4	
UXSR	4	
VINS	4	
PACX		2
except for		
EXPU		3
RMEN		3
EXLI		3
requests		3
(CPSN form.)		
ISEP	2	
ISOS	2	
EMLD	2	
EMUP	2	
CPSN	3	
EMSN		3
MESN	4	
SASN	4	
ACTI	3	
PQCE		2
GETA		2
GETD		2

PROCEDURE	GLOBAL AUTHORIZATION	DATABASE AUTHORIZATION
RVDE		2
RVKE		2
XPAF		2
XPDM		2
PRGS		2
CSES	4	
ESES	4	
GRPE	4	
INPE	4	
PRPE		2
RSPE	4	
SIPE		3
SVPE	4	
TRJC	4	
TRUP	4	
TRDU	4	
TRPF	4	
TRRP	4	
TRRT	4	
VDWN	4	
VUP1	4	
VUP2	4	
VPUR	4	

For platforms that do not support Database authorizations, do not take the two authorization types into account.

For platforms supporting database authorizations, when this level is not specified, the system performs the check on the global authorization level.

The following procedures do not require an authorization access check:

UPDT, UPDP, HIPE, and GPRT: standard Database access check.

PARM, LOAE, and CRYP: authorization for parameters update.

1.3. ABNORMAL ENDINGS

ABNORMAL EXECUTIONS

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

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

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

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

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

2. DATABASE MANAGEMENT UTILITIES

2.1. MLIB: DATABASE MANAGEMENT

2.1.1. MLIB: INTRODUCTION

MLIB: INTRODUCTION

The Database Management (MLIB) procedure has a two-fold purpose:

. Initialize the database in the form of a sequential file (or 2 files if the Dispatch option is used), called 'PC', which is then used as input to the Restoration (REST) procedure.

. Create or delete libraries in an existing database.

EXECUTION CONDITIONS

The database must be closed to on-line access and use, unless the current execution is a simulation. The MLIB procedure must be followed by the REST procedure so that the new library structure is taken into account.

Batch procedure authorization access option: Global authorization level 4 is required.

ABENDS

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

2.1.2. MLIB: INPUT - PROCESSING - RESULTS

MLIB : INPUT-PROCESSING-RESULTS

USER INPUT

Batch procedure authorization access option:
One '*' line with user code and password.

There are two types of specific user input:

. Heading line (required) at the top of the input file that specifies a new database to be initialized or an existing database to be retrieved.

. As many lines (optional) as there are libraries to be created, modified or deleted.

The structure of the heading line is as follows:

POS.	LEN.	VALUE	MEANING
2	1	'G'	Line code
3	1	' '	Modification of existing database
		'I'	Initialization of new database
4	1	' '	Actual update
		'S'	Simulated update

Update simulation is used to obtain the state of the database as it would appear if the requested modifications had actually been implemented.

It allows the user to judge the impact of a change in the structure of the database before actual execution. For large databases, actual execution may use a lot of machine time.

The structure of the 'library' lines is as follows:

POS.	LEN.	VALUE	MEANING
1	1	'C'	Creation
		'M'	Modification
		'D'	Deletion
2	1	'*'	Line code
3	3	bbb	Code of the library to update
6	3	ccc	Code of the upper level library

NOTE: Asterisks ("*") cannot be used in the library codes because they are not compatible with the WorkStation.

UPDATE RULES

Updates are executed line by line. No previous transaction sort is executed. The resulting database must remain consistent during the update.

1. DELETION TRANSACTIONS:

A library with dependent libraries cannot be deleted. To delete an entire sub-network, begin by deleting the libraries at the lowest hierarchical level and work upward to the highest level.

The upper library code must not be entered on library deletion lines. Only the code of the library to be deleted may be specified.

The deletion of a library causes this library's entire contents to be deleted. Its contents are replaced by empty records, or 'gaps'. (See the REST restoration procedure.)

2. CREATION TRANSACTIONS:

When a library is created, it can only be linked to an already existing library or to a library that was previously created in the update job stream.

Therefore, always create the 'parent' library before its 'child' libraries. Both can however be created by the same run of the procedure.

Note: A VisualAge Pacbase Database cannot contain more than 300 libraries.

3. MODIFICATION TRANSACTIONS:

Generally, transactions modify links between libraries. This modification often involves inserting a new library between two existing libraries. The new library, which must be empty, becomes the 'central' library of the library at the lower hierarchical level. This new 'central' library must be attached directly or indirectly to the former 'central' library.

Structure loops are detected by the system.

A library may not be deleted and re-created during the same run.

When an error is detected on a line, a message is generated, and the update is interrupted because the resulting database would otherwise be inconsistent. The line containing the error must be corrected and the job restarted, as the initial database will not have been modified.

PRINTED REPORTS

In all cases, a report on the initial state of the database and an update report are printed.

If no errors have been detected, a report on the database is printed after the update.

RESULTS

If no errors are detected and if the update is 'real' (not simulated), the result is a sequential image of the updated database (PC), which serves as input for database reloading.

WARNING

This procedure does not allow for the recovery of disk space when libraries are deleted. Records are physically present in the database as 'gaps'. It is the Reorganization (REOR) procedure that deletes these gaps so that disk space can be recovered.

This procedure increments the session number.

2.1.3. MLIB: DESCRIPTION OF STEPS

MLIB: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE VALIDATION: PTU100

This program is always executed.

```
.Permanent input files:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN  :    DSN=&INDEX..&ROOT.&FILE.AN
 -Printing-generation request file
  (in input-output if no simulation)
  PAC7AG  :    DSN=&INDEX..&ROOT.&FILE.AG
 -Error message file
  (in input-output)
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Input transaction file:
 -Update transactions
  PAC7MB:  DSN=&&MLIBMB

.Output files:
 -Sequential image of data
  PAC7RP:  DSN=&&MLIBPR  Length=149
  (must have capacity for all data)
 -Sequential image of indexes
  PAC7NA:  DSN=&&MLIBAN  Length=54
  (must have capacity for all indexes)
 -Sequential image of unsorted indexes
  PAC7NB:  DSN=&&MLIBNB  Length=54
 -Temporary storage
  PAC7RQ:  DSN=&&MLIBPQ  Length=149
  (1 record)
```

```
.Output reports:
 -List of user transactions
  PAC7EV
 -Report on database before and after
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

When the database is initialized, only the after-image
is printed.

.Return codes:
 - 0:  OK without simulation,
 - 4:  OK with simulation,
 - 8:  No batch-procedure access authorization
       Error on input transactions,
 -12:  Error in accessing database. In this case, an ABEND
       is provoked and a DUMP is printed.
```

Note: The database files AN, AR, and AG are not open during the database initialization procedure.

SEQUENTIAL-IMAGE FORMATTING: PTU120

```
This program is executed only when there is no simulation
and when there are no errors on the input transactions.

.Internal sort files
 SORTWK01
 SORTWK02
 SORTWK03

.Permanent input files:
 -Data file
  (in input-output to update session number)
  PAC7AR  :     DSN=&INDEX..&ROOT.&FILE.AR

.Temporary files:
 -The 4 output files from the preceding step.

.Output file:
 -Sequential image of the database
  PAC7PC  :    DSN=&INDEXQ..&ROOT.&FILE.PC(+1)
 If Dispatch backup option:
 -Database sequential image 2
  PAC7PD  :    DSN=&INDEXQ..&ROOT.&FILE.PD(+1)

.Output reports:
 -None.
```

2.1.4. MLIB: EXECUTION JCL

```
//*********************************************************************
//*  VA PAC       : LIBRARY MANAGEMENT                                  *
//*********************************************************************
//$RADP.MLIB PROC FILE=$FILE,        PHYSICAL-DATABASE NUMBER
//           ROOT=$ROOT,             ROOT OF THE VA PAC SYSTEM
//           INDEX='$INDEX',         VSAM INDEX
//*:         VSAMCAT='$CATU',        VSAM USER CATALOG
//*:         SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//           OUT='$OUT',             OUTPUT CLASS
//           VOLS='SER=$VOLO',       VOLUME OF BACKUP FILE
//           UNITS=$UNITO,           VOLUME OF BACKUP FILE
//           INDEXP='$INDEXP',       INDEX OF NON VSAM FILES
//           INDEXQ='$INDEXQ',       INDEX OF DATA GROUP FILES
//           SPAAN='(TRK,(050,10),RLSE)',   TEMPORARY INDEX SPACE
//           SPAAR='(TRK,(050,10),RLSE'),   TEMPORARY DATA SPACE
//           SPAPC='(TRK,(100,10),RLSE)',   MODIFIED DATABASE SPACE
//           STEPLIB='$MODB',        LIBRARY OF BATCH LOAD-MODULES
//           PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//           DBDLIB='$DBDLIB',       LIBRARY OD DBD'S
//           RESLIB='$RESLIB',       RESLIB IMS
//           PROCLIB='$PRCLIB',      PROCLIB IMS
//           SORTLIB='$BIBT',        SORT LIBRARY
//           UWK=$UWK,               WORK UNIT
//           CYL=1,                  SORTWORK SPACE
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//* INPUTS :                                                          *
//*                                                                   *
//* HEADING LINE  (REQUIRED) :                                        *
//* COL  2     : 'G'                                                  *
//* COL  3     : 'I' TO INITIALIZE A NEW DATBASE, BLANK OTHERWISE     *
//* COL  4     : 'S' TO EXECUTE A SIMULATION, BLANK FOR REAL UPDATE.  *
//*                                                                   *
//* CURRENT LINE:(ONE PER MODIFICATION OF THE DATABASE) :             *
//* COL  1      : ACTION CODE(C=CREATION, M=MODIFICATION, D=DELETION) *
//* COL  2      : '*'                                                 *
//* COL  3 A  5 : CODE OF THE LIBRARY TO BE CREATED OR                *
//*               CODE OF THE LIBRARY TO BE DELETED OR                *
//*             CODE OF THE LIBR. WHOSE 'CENTRAL' LIBR. IS TO BE MOD. *
//* COL  6 A  8 : CODE OF THE 'CENTRAL' LIBRARY                       *
//*                                                                   *
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&MLIBMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAG     DD DSN=&INDEX..&ROOT.&FILE.AG,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAG),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PTU100  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU100,PTU100$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
```

```
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AG$SUF DD DSN=&INDEX..&ROOT.&FILE.AG,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7MB   DD DSN=&&MLIBMB,DISP=(OLD,DELETE)
//PAC7NA   DD DSN=&&MLIBAN,DISP=(,PASS,DELETE),
//            UNIT=&UWK,SPACE=&SPAAN,
//            DCB=(RECFM=FB,LRECL=54,BLKSIZE=27270)
//PAC7NB   DD DSN=&&MLIBNB,DISP=(,PASS,DELETE),
//            UNIT=&UWK,SPACE=(TRK,(10,1),RLSE),
//            DCB=(RECFM=FB,LRECL=54,BLKSIZE=6264)
//PAC7RP   DD DSN=&&MLIBPR,DISP=(,PASS,DELETE),
//            UNIT=&UWK,SPACE=&SPAAR,
//            DCB=(RECFM=FB,LRECL=149,BLKSIZE=27267)
//PAC7RQ   DD DSN=&&MLIBPQ,DISP=(,PASS,DELETE),
//            UNIT=&UWK,SPACE=(TRK,1,RLSE),
//            DCB=(RECFM=FB,LRECL=149,BLKSIZE=149)
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//PAC7EV   DD SYSOUT=&OUT
//*
//PTU120  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU120,PTU120$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(0,NE,PTU100)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7AN   DD DSN=&&MLIBAN,DISP=(OLD,PASS)
//PAC7NB   DD DSN=&&MLIBNB,DISP=(OLD,PASS)
//PAC7PR   DD DSN=&&MLIBPR,DISP=(OLD,PASS)
//PAC7PQ   DD DSN=&&MLIBPQ,DISP=(OLD,PASS)
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAPC,
```

```
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7PD   DD DSN=&INDEXQ..&ROOT.&FILE.PD(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//*
```

2.2. SAVE: DATABASE BACKUP

2.2.1. SAVE: INTRODUCTION

SAVE: INTRODUCTION

The purpose of the Database Backup procedure (SAVE) is to format sequentially the main files that make up the database. The resulting files have the 'PC' format.

The back-up is performed on the following files:

. Data file (AR),
. Index file (AN).

An option allows for a database backup in two sequential files: one for the data (backup of the AR file), one for the indices (backup of the AN file).

This option (DISPACTH or NO DISPATCH) is implemented in the database restoration procedure. For further details, see the REST procedure user input description.

EXECUTION CONDITIONS

On-line access must be prohibited in order to preserve database integrity during execution of the SAVE procedure.

Batch procedure authorization access option: global authorization level 4 is required.

ABENDS

Refer to Chapter 'OVERVIEW', Subchapter 'ABNORMAL ENDINGS'.

The most common cause for an abend in the SAVE procedure is that the on-line environment is still open to transactions. The procedure can therefore be restarted once the on-line environment is closed.

ARCHIVAL AND BACKUP LINKING

If the backup procedure is preceded by a Journal archival (ARCH procedure), its execution may be conditionned by the return code of the PTU320 ARCH step, i.e.:

. 0: No error detected

. 8: Database not available

SIMPLIFIED BACKUP

Files may also be backed up via standard system utilities. In this case, run the SASY procedure to check the consistency of data and indexes. (See Sub-chapter 'Database system backup.)

2.2.2. SAVE: PROCESSING - RESULTS

SAVE: INPUT-RESULTS

PRINTED REPORT

Once the SAVE procedure is executed, the following reports are printed:

- A report containing the number of records saved in each file, and the session number

- Two optional reports:

. a statistical report with number of records per library and per line-type

. a limitation report (listing database limits reached, such as the number of calls to the same macro-structure).

USER INPUT

Batch-procedure access authorization option:
One '*' line with user code and password.

The user may cancel the formatting and the output of statistical reports on the database, in order to speed up the execution of the SAVE procedure.

If a cancellation request is not made, all reports will be printed.

The structure of the line is as follows:

POS.	LEN.	VALUE	MEANING
2	2	'OR'	LINE CODE
8	1		STATISTICAL REPORT BY LIBRARY OF THE DATABASE THAT HAS BEEN BACKED UP
		' '	PRINTING OF STATISTICS
		'N'	NO PRINTING OF STATISTICS
9	1		REPORT INDICATING THE P.M.S. CALL LIMITATIONS IN THE DATABASE
		' '	PRINTING OF LIMITATIONS
		'N'	NO PRINTING OF LIMITATIONS

OUTPUT

The output of the SAVE procedure is the following:

. Either a unique sequential file (PC), of variable length, containing the mirror of the two saved files,

. Or two sequential files, one of variable length containing the mirror of the data (PC), the other of fixed length containing the mirror of indices (its name depends on the platform).

If the database is no longer consistent after an abend during the last update, the SAVE procedure will not be executed.

If the database is inconsistent, the procedure sends back a return code.

NOTES:

The SAVE procedure increments the current session number.

The Generation-Print Request file (AG) is not saved by this procedure. A special procedure (SVAG) does it. (See Chapter 'SVAG: GENERATION-PRINT REQUEST BACKUP.)

2.2.3. SAVE: DESCRIPTION OF STEPS

SAVE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE CONSISTENCY CHECK: PTUBAS

```
.Permanent input files:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Output report
 -Validity report (Length=079)
  PAC7DS

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

BACKUP OF THE DATABASE: PTU500

```
.Permanent input then input-output file:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR

.Permanent input files:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE
 -Index File
  PAC7AN  :    DSN=&INDEX..&ROOT.&FILE.AN

.Input transaction file:
 -User transaction
  PAC7MB:  DSN=&&SAVEMB

.Output file:
 -Sequential image of the database
  PAC7PC  :    DSN=&INDEXQ..&ROOT.&FILE.PC(+1)
 If backup Dispatch option:
 -Sequential image 2 of the database
  PAC7PD  :    DSN=&INDEXQ..&ROOT.&FILE.PD(+1)

.Output reports:
 -Backup review
  PAC7EU
 -Statistics on database
  PAC7DS
 -Batch-procedure authorization option
  PAC7DD

Return code(s):

.8: no batch procedure authorization option or
    inconsistent database
```

Response to return code:

This program sends the return code '8' in case of database inconsistency. The backup is then deleted by the next step in the procedure and a restoration must be performed using the last valid backup.

If there is no other backup, the user should first contact VisualAge Pacbase Support. Then, the inconsistent database should be saved by the same procedure with the backup deletion step inactive. The resulting backup contains only data, and can only be used after running the REOR procedure.

BACKUP DELETION IN CASE OF DATABASE INCOHERENCE: IEFBR14

```
.Input file:
 -Database sequential image
  DDPC  :    DSN=&INDEXQ..&ROOT.&FILE.PC(+1)
 If backup Dispatch option:
 -Database sequential image ¾2
  DDPD  :    DSN=&INDEXQ..&ROOT.&FILE.PD(+1)

This step is executed if the return code of the previous
program is other than 0.
```

2.2.4. SAVE: EXECUTION JCL

```
//*********************************************************************
//*  VA PAC       : BACKUP OF THE DATABASE                            *
//*********************************************************************
//$RADP.SAVE PROC FILE=$FILE,   PHYSICAL-DATABASE NUMBER
//          ROOT=$ROOT,         ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',     VSAM INDEX
//*:        SYSTCAT='$CATV',    VSAM SYSTEM CATALOG
//*:        VSAMCAT='$CATU',    VSAM USER CATALOG
//          OUT='$OUT',         OUTPUT CLASS
//          VOLS='SER=$VOLO',   BACKUP VOLUME
//          UNITS='$UNITO',     BACKUP UNIT
//          INDEXP='$INDEXP',   NON VSAM FILE INDEX
//          INDEXQ='$INDEXQ',   DATA GROUP FILE INDEX
//          SPAPC='(CYL,(3,1),RLSE)',  BACKUP SPACE
//          UWK='$UWK',         WORK UNIT
//          STEPLIB='$MODB',    LIBRARY OF BATCH LOAD-MODULES
//          PSBLIB='$PSBLIB',   LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',   LIBRARY OF DBD'S
//          RESLIB='$RESLIB',   IMS RESLIB
//          PROCLIB='$PRCLIB',  IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//*  INPUT FOR OPTIONAL REPORTS                                       *
//*  COL 02     : 'OR'                                                *
//*  COL 08     : ' ' --> VA PAC STATISTICS PRINTING                  *
//*             : 'N' --> NO VA PAC STATISTICS PRINTING               *
//*  COL 09     : ' ' --> VA PAC LIMITATIONS PRINTING                 *
//*             : 'N' --> NO VA PAC LIMITATIONS PRINTING              *
//*********************************************************************
//*
//COPY    EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&SAVEMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PTUBAS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUBAS,PTUBAS$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
```

```
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DS   DD SYSOUT=&OUT
//*
//PTU500  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU500,PTU500$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTUBAS)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7MB   DD DSN=&&SAVEMB,DISP=(OLD,PASS)
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7PD   DD DSN=&INDEXQ..&ROOT.&FILE.PD(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7DD   DD SYSOUT=&OUT
//PAC7DS   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//*
//DELPC  EXEC PGM=IEFBR14,COND=(8,NE,PTU500)
//*****************************************
//DDPC     DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),DISP=(OLD,DELETE)
//DDPD     DD DSN=&INDEXQ..&ROOT.&FILE.PD(+1),DISP=(OLD,DELETE)
```

2.3. SASY: DATABASE SYSTEM BACKUP COMPLEMENT

2.3.1. SASY: INTRODUCTION

SASY : INTRODUCTION

The Database System Backup Complement procedure (SASY) allows you to save the Database using any Operating System's utility, while at the same time creating a checkpoints, through an increment of the session number.

The following files are to be backed up:

. Data file (AR),

. Index file (AN).

EXECUTION CONDITIONS

The on-site database backup utility must have been executed on the Data (AR) and Index (AN) files.

The transaction Journal file (AJ) must have been archived via the ARCH procedure.

The database must be closed to on-line use in order to maintain its consistency during the backup.

ABEND

The main cause of an abend is that the database remained open to on-line use while the procedure was executing.

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

USER INPUT

No user input is necessary when requesting execution of the SASY procedure.

RESULT

This procedure increments the current session number.

If the database is in an inconsistent state due to an abend in the last update, the SASY procedure is not executed and the backup executed by the on-site Operating System utility is not valid.

2.3.2. SASY: DESCRIPTION OF STEPS

SASY: DESCRIPTION OF STEPS

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE CONSISTENCY CHECK: PTUBAS

```
.Permanent input files:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Output report
 -Validity report (Length=079)
  PAC7DS

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

SESSION NUMBER INCREMENTATION: PTU502

```
.Permanent input-output file:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR

.Permanent input file:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Output Report:
 -Review
  PAC7GZ
```

2.3.3. SASY: EXECUTION JCL

```
//*********************************************************************
//*  VA PAC       :  SYSTEM BACKUP     (IBM)                          *
//*********************************************************************
//$RADP.SASY PROC FILE=$FILE,    PHYSICAL-DATABASE NUMBER
//           ROOT=$ROOT,         ROOT OF THE VA PAC SYSTEM
//           INDEX='$INDEX',     VSAM INDEX
//           INDEXP='$INDEXP',   NON VSAM FILE INDEX
//           STEPLIB='$MODB',    LIBRARY OF BATCH LOAD-MODULES
//*:         SYSTCAT='$CATV',    VSAM SYSTEM CATALOG
//           OUT='$OUT',         OUTPUT CLASS
//           RESLIB='$RESLIB',   IMS RESLIB
//           PROCLIB='$PRCLIB',  IMS PROCLIB
//           DBDLIB='$DBDLIB',   LIBRARY OF DBD'S
//           PSBLIB='$PSBLIB',   LIBRARY OF PSB'S
//           BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//           CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//PTUBAS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUBAS,PTUBAS$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DS   DD SYSOUT=&OUT
//*
//*--------------------------------------------------------------------*
//*  INSERT HERE THE SYSTEM BACKUP STEP(S) OF DATABASES 'AN' & 'AR'  *
//*                                                                    *
//*  IMPORTANT: THE STEP(S) MUST BE EXECUTED ONLY IF:                  *
//*             COND=(00,NE,PTUBAS)                                    *
//*--------------------------------------------------------------------*
//*
//PTU502  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU502,PTU502$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTUBAS)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
```

```
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7GZ   DD SYSOUT=&OUT
//*
```

2.4. REST: DATABASE RESTORATION

2.4.1. REST: INTRODUCTION

REST: INTRODUCTION

The Database Restoration procedure (REST) re-creates a database that can be manipulated on-line, using the sequential image produced by the Back-up (SAVE), the Database Management (MLIB), the Reorganization (REOR, QREO) and Storage Optimization of Multi-volume Data (STOP) procedures.

It also allows both the retrieval of archived transactions and the modification of the number of gaps in the database.

EXECUTION CONDITIONS

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 Parameter library.

The REST procedure physically and logically reinitializes the Journal file, which must have been saved previously by the ARCH procedure.

Batch procedure access authorization option: global authorization level 4 is required.

ABNORMAL EXECUTION

Refer to chapter 'OVERVIEW', subchapter 'ABNORMAL ENDINGS'.

Regardless of the cause, the procedure can be restarted as it is once the problem is solved.

CHECKPOINT REQUEST

This facility allows you to request synchronization points during a batch update (UPDT procedure) or during a database restoration (REST or RESY procedures).
In case of ABEND, a ROLLBACK is performed, thus securing a coherent database.

Therefore, it is always possible, after an abnormal ending of the UPDT procedure, to restart the procedure without executing a restoration. However, it is recommended to delete transactions already taken into account.

Checkpoints are performed at a frequency rate defined by the user.

EXAMPLE: A '0100' frequency rate means that a checkpoint is performed every 100 transactions.

INPUT OF THE CHECKPOINT FREQUENCY RATE FOR A BATCH UPDATE

The checkpoint frequency rate is entered on a single 'Y'-line located BEFORE the first '*'-line. The 'Y'-line is formatted as follows:

POSITION	LENGTH	VALUE	MEANING
2	1	Y	LINE CODE
4	4	nnnn	CHECKPOINT FREQUENCY RATE (DEFAULT VALUE=0000)

For the REST and RESY procedures, the checkpoint frequency is entered in the User Input.

2.4.2. REST: USER INPUT

REST : USER INPUT

Batch procedure access authorization: one '*' line with user code and password.

The structure of the specific input is described in the chart below.

POS.	LEN.	VALUE	MEANING
2	1	Y	Line code
3	5	nnnnn	Number of unused gaps
8	2	pp	Number of unused gaps as a percentage
10	1	F	French
		E	English
11	1	0	No suppression of journal
		1	Suppression of journal (no journali zation of update transactions)
		blank	Previous value
12	1		This field may only be used with DOS/VSE
		I	Default option for all hardware (1)
		N	DOS/VSE: if CURRENT-DATE = DD/MM/YY
13	3	REC	If archived transactions are recov'd.
16	4	XXXX	4-character Database code chosen by the Database Manager (displayed in the top-right corner of VA Pac screens) DATABASE CODE IS REQUIRED
20	3	nnn	Maximum access number: on-line search (lists) (default value: 300)
23	1	U	Implicit update (default option)
		N	Explicit update
24	4	nnnn	Checkpoint frequency (IMS, UNISYS, GCOS7, and GCOS8 only) if REC in col. 13 (default: nnnn=0000)
28	7		Not used.
35	12		PFkeys assigned functions (2).
79	1		Dispatch option of Backup:
		'D'	Dispatch: sequential back-up of the database in two separate files.
		'N'	No Dispatch: standard backup of the database in one PC file.
		' '	Same as previous restoration.

When there is no input, the database characteristics remain unchanged. The default language option is French. Any area left blank will default to current option selections.

The user can insert 'gaps' into the database (empty records to be used to create new data).

(1): This date is used:

. For documentation printing purposes
. To check the system expiration date
. For transaction archiving.

Accidentally setting this date to 'N' may cause problems, such as making it impossible to select archived transactions by date (EXPJ), or even to use the Database, in which case the following message is displayed:

"SYSTEM EXPIRATION DATE".

It is important to check that this indicator is set correctly in each Database.

(2): 12-position table, with each position referring to a standard function.

To modify the PFkey assigned to a function, the value of the new PFkey coded in base 36 is entered in the corresponding position in the table.
For example, to assign function 1 to PFKey 17, enter code 'H' in position 1 of the table.

No validation procedure is executed by the system. The PFkey assignment may be viewed on the corresponding sub-menu.

NOTES:

(Gaps do not apply to IMS, GCOS8, OS/2, UNIX or WINDOWS/NT Databases.)

- The number of gaps entered is the minimum number for the database. If the database already contains more gaps than the number requested on input, this transaction will have no effect on the database. If the number of gaps in the database is smaller, the number of gaps allowed will be increased.

- A number of gaps equal to NULL does not prevent the update of the Database, but reduces its performance.

- The limit of on-line accesses to the Journal depends on the number specified as input of the restoration procedure.

If you do not want the update transactions of the database to be saved in the Journal file, you can turn the 'journalization' off by setting this parameter to '1'. In this case, it is not possible to restore the database using the recovery of archived transactions ('REC' entered on the input parameter card). It is therefore highly recommended to set this parameter to 0 (which is the default option), in order to avoid restoration problems.

In case of error, invalid parameters are ignored, and the system ensures restoration using the parameter values stored in the sequential image of the database.

SIMPLIFIED RESTORATION

If the backup was performed via a system utility followed by the SASY procedure, restoration via a utility must be followed by the RESY procedure, which ensures the consistency between files.

OUTPUT REPORTS

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

GENERAL RESULTS

Once the procedure has been executed, the database is ready to be used in batch or on-line mode.

Even if the resulting database contains no gaps, it is still possible to do an update. To do this, the system takes advantage of the features of the access method in use, which may have a negative effect on system performance.

Therefore, it is highly advisable to secure a sufficient number of gaps in the database in order to optimize system performance, thus avoiding sometimes costly updates when using access methods for space management.

NOTE: Once this procedure is executed, the current session number is the same as the session number of the sequential image, or of the most recent transaction, if you've requested archived transaction retrieval.

2.4.3. REST: DESCRIPTION OF STEPS

REST: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

```
.Input file:
  CARTE

.Output file:
  PAC7MB:  DSN=&&MB

.Permanent input file:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Output report:
 -Batch-procedure authorization option:
  PAC7DD

.Return code(s):
 - 8:  No batch-procedure authorization.
```

VALIDATION OF JOURNAL EXISTENCE: IDCAMS

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

```
    .0    :  The Journal file exists
    .OTHER:  The Journal file does not exist
```

VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed only if the Journal file exists.

```
.Permanent input files:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE
 -Journal file
  PAC7AJ  :    DSN=&INDEX..&ROOT.&FILE.AJ

.Output report:
  PAC7EU
  It is printed if the Journal file was not archived.

.Return codes:
 - 0:  The Journal file has been archived.
 - 8:  The Journal file was not archived. In this case, no
       other steps are executed.
```

DEFINITION OF FILES: IDCAMS

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

```
.Defined files:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Journal file
  PAC7AJ:  DSN=&INDEX..&ROOT.&FILE.AJ
```

RESTORATION OF THE DATABASE: PTU400

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

```
.Permanent input files:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE
 -Sequential image of the database
  PAC7PC  :    DSN=&INDEXQ..&ROOT.&FILE.PC(0)
 If backup option Dispatch:
 -Sequential image of database ¾2
  PAC7PD  :    DSN=&INDEXQ..&ROOT.&FILE.PD(0)

.Permanent output files:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR
 -Index File
  PAC7AN  :    DSN=&INDEX..&ROOT.&FILE.AN
 -Journal file
  PAC7AJ  :    DSN=&INDEX..&ROOT.&FILE.AJ

.Input transaction file:
 -User transactions
  PAC7MB:  DSN=&&RESTMB

.Output file:
 -Working file (2 records)
  PAC7PS:  DSN=&&PAC7PS

.Output reports:
 -Restoration report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD
```

DATABASE AVAILABILITY - TRANSACTION RETRIEVAL: PTU420

This step is executed if the Journal file has been archived. It retrieves the appropriate transactions and executes an update on the first record of the Data file. It is REQUIRED for a coherent database.

```
.Input-output file:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR

.Permanent input files:
 -Journal to apply
  PAC7JO  :    DSN=&INDEXQ..&ROOT.&FILE.PJ(0)
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Input work file:
  PAC7PS:  DSN=&&PAC7PS

.Output file:
 -Update transactions
  PAC7OJ:  DSN=&&PAC7OJ  Length=167

.Output report:
 -Retrieval report
  PAC7EU

.Return code(s):
 - 0:  There are transactions to retrieve.
 - 4:  No transactions to retrieve
       OR erroneous user input.
```

In case of an abnormal end in this step, the database cannot be updated.

DATABASE UPDATE: PACA15

.Permanent update files:
 -Data file
  PAC7AR: DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN: DSN=&INDEX..&ROOT.&FILE.AN
 -Journal file
  PAC7AJ: DSN=&INDEX..&ROOT.&FILE.AJ

.Permanent input files:
 -Error message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE
 -DSMS file of VA Pac elements
  PAC7D3: DSN=&INDEXD..&ROOTD.&FILED.D3
  PAC7DC: DSN=&INDEXD..&ROOTD.&FILED.DC
  (DSM variant only)

.Input transaction file:
 -Update transactions
  PAC7MV: DSN=&&PAC7MV

.Output report(s):
 -Update report
  PAC7IE (Length=132)
 -Erroneous-transaction list
  PAC7IF (Length=132)
 (The list of transactions belonging to a user is preceded
 by a banner specifying the user code.)

.Return code(s):
 -0: OK, no error
 -2: Warning
 -4: Critical error

2.4.4. REST: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : RELOADING-RESTORATION OF THE DATABASE             *
//*********************************************************************
//$RADP.REST PROC FILE=$FILE,       PHYSICAL-DATABASE NUMBER
//          ROOT=$ROOT,             ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',         VSAM INDEX
===SEQ FOR DSM
//          ROOTD=$ROOTD,           ROOT OF THE DSMS SYSTEM
//          FILED=$FILED,           DSMS PHYSICAL DATABASE NUMBER
//          INDEXD='$INDEXD',       DSMS SYSTEM VSAM INDEX
===SEQ
//*:        VSAMCAT='$CATU',        VSAM USER CATALOG
//*:        SYSTCAT='$CATV',        VSAM SYSTEM CATLOG
//          OUT='$OUT',             OUTPUT CLASS
//          INDEXP='$INDEXP',       NON-VSAM FILE INDEX
//          INDEXQ='$INDEXQ',       DATA GROUP FILE INDEX
//          SPAOJ='(TRK,(010,05),RLSE)',   JOURNAL SPACE
)//          SPAAJ='(CYL,(3,2),RLSE)',      AJ FILE SPACE (OSAM DB
//          SPAAR='(CYL,(6,2),RLSE)',  AR FILE SPACE (OSAM DATABASE)
//          VOLU='$VOLU',           AJ & AR DATABASES VOLUME
//          UNITU='$UNITU',         AJ & AR DATABASES UNIT
//          STEPLIB='$MODB',        LIBRARY OF BATCH LOAD-MODULES
//          PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//          RESLIB='$RESLIB',       IMS RESLIB
//          PROCLIB='$PRCLIB',      IMS PROCLIB
//          UWK=$UWK,               WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//*  INPUT                                                            *
//*  COL 02     : 'Y'                                                 *
//*  COL 03-07  : NUMBER OF GAPS IN ABSOLUTE VALUE                    *
//*  COL 08-09  : NUMBER OF GAPS IN PERCENTAGE ( / DATABASE )         *
//*  COL 10     : INITIAL LANGUAGE CODE ( F=FRENCH, E=ENGLISH)        *
//*  COL 11     : '1' TO INHIBIT THE TRANSACTION JOURNAL              *
//*  COL 12     : FORMAT OF MACHINE DATE    ('N' FOR DD/MM/YY)        *
//*                                         ('I' FOR MM/DD/YY)        *
//*  COL 13-15  : 'REC' TO REQUEST RETRIEVAL OF ARCHIVED TRANSACTIONS *
//*  COL 16-19  : 4 CHARACTERS TO BE DISPLAYED ON ALL VA PAC SCREENS  *
//*  COL 20-22  : 'NNN' MAXIMUM NUMBER OF SEARCH ACCESSES TO THE      *
//*               DATABASE (LISTS) - (DEFAULT VALUE: 300)             *
//*  COL 23     : 'U' (DEFAULT VALUE).: IMPLICIT UPDATE               *
//*               'N' .................: EXPLICIT UPDATE             *
//*  COL 24-27  : CHECKPOINT FREQUENCY (IMS ONLY)                     *
//*               IF REC IN COL.13: DEFAULT = 0000                    *
//*  COL 28-33  : RESERVED FOR PACDESIGN (NOT USED)                   *
//*  COL 34     : NOT USED                                            *
//*  COL 35-46  : PF-KEYS SIGNIFICATION    (1 -> 9  +  A -> O)        *
//*  COL 79     : BACKUP FILES DISPATCH                               *
//*             : 'N' (DEFAULT VALUE) : NO DISPATCH (1 FILE)          *
//*             : 'D' : DISPATCH (2 FILES)                            *
//*                                                                   *
//*  IN CASE OF NO INPUT, RELOADING DOES NOT MODIFY THE NUMBER OF     *
//*  EXISTING GAPS, AND THE REST OF THE DATA IS NOT MODIFIED          *
//*                                                                   *
//*  IF THE JOURNAL DATABASE OF TRANSACTIONS ON DISK (AJ) IS NOT      *
//*  REINITIALIZED, THE RESTORE CHAIN IS NOT EXECUTED.                *
//*  IT IS THEREFORE NECESSARY TO EXECUTE THE 'ARCH' PROCEDURE FIRST. *
//*                                                                   *
//*********************************************************************
//*
//PTU004  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU004,PTU004$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
```

```
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//PAC7DD   DD SYSOUT=&OUT
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//EXISAJ  EXEC PGM=IDCAMS,COND=(00,NE,PTU004)
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(LI&ROOT.&FILE.AJ),DISP=SHR
//*
//CONTAJ  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU380,PTU380$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=((00,NE,PTU004),(00,NE,EXISAJ))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,DISP=SHR
//PAC7EU   DD SYSOUT=&OUT
//*
//DEFINE  EXEC PGM=IDCAMS,COND=((00,NE,PTU004),(00,NE,CONTAJ))
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.AN),DISP=SHR
//*
//OSAMAJ1 EXEC PGM=IDCAMS,COND=((00,NE,PTU004),(00,NE,CONTAJ))
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DL&ROOT.&FILE.AJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
```

```
    //*
    //OSAMAJ2 EXEC PGM=IEFBR14,COND=((00,NE,PTU004),(00,NE,CONTAJ))
    //PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,UNIT=&UNITU,
    //            DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
    //            DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
    //            SPACE=&SPAAJ
    //*
    //OSAMAR1 EXEC PGM=IDCAMS,COND=((00,NE,PTU004),(00,NE,CONTAJ))
    //SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DL&ROOT.&FILE.AR),DISP=SHR
    //SYSPRINT DD SYSOUT=&OUT
    //*
    //OSAMAR2 EXEC PGM=IEFBR14,COND=((00,NE,PTU004),(00,NE,CONTAJ))
    //PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,UNIT=&UNITU,
    //            DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
    //            DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
    //            SPACE=&SPAAR
    //*
    //PTU400  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
    //        PARM=(DLI,PTU400,PTU400$SUG,&BUF,
    //        &SPIE&TEST&EXCPVR&RST,&PRLD,
    //        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
    //        COND=((00,NE,PTU004),(00,NE,CONTAJ))
    //STEPLIB  DD DSN=&RESLIB,DISP=SHR
    //         DD DSN=&STEPLIB,DISP=SHR
    //DFSRESLB DD DSN=&RESLIB,DISP=SHR
    //IMS      DD DSN=&PSBLIB,DISP=SHR
    //         DD DSN=&DBDLIB,DISP=SHR
    //*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
    //*:       DD DSN=&VSAMCAT,DISP=SHR
    //SYSOUT   DD SYSOUT=&OUT
    //SYSOUX   DD SYSOUT=&OUT
    //DDSNAP   DD SYSOUT=&OUT
    //PROCLIB  DD DSN=&PROCLIB,DISP=SHR
    //IEFRDER  DD DUMMY,
    //            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
    //SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
    //            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
    //IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
    //            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
    //IMSMON   DD DUMMY
    //DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
    //PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
    //PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,DISP=SHR
    //PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
    //PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
    //PAC7DD   DD SYSOUT=&OUT
    //PAC7EU   DD SYSOUT=&OUT
    //PAC7MB   DD DSN=&&MB,DISP=(OLD,DELETE)
    //PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(0),DISP=OLD
    //PAC7PD   DD DSN=&INDEXQ..&ROOT.&FILE.PD(0),DISP=OLD
    //PAC7PS   DD DSN=&&PAC7PS,DISP=(,PASS),UNIT=&UWK,
    //            DCB=(RECFM=FB,LRECL=140,BLKSIZE=6160),
    //            SPACE=(TRK,1)
    //*
    //PTU420  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
    //        PARM=(DLI,PTU420,PTU420$SUG,&BUF,
    //        &SPIE&TEST&EXCPVR&RST,&PRLD,
    //        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
    //        COND=((00,NE,PTU004),(00,NE,CONTAJ),(00,NE,PTU400))
    //STEPLIB  DD DSN=&RESLIB,DISP=SHR
    //         DD DSN=&STEPLIB,DISP=SHR
    //DFSRESLB DD DSN=&RESLIB,DISP=SHR
    //IMS      DD DSN=&PSBLIB,DISP=SHR
    //         DD DSN=&DBDLIB,DISP=SHR
    //*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
    //SYSOUT   DD SYSOUT=&OUT
    //SYSOUX   DD SYSOUT=&OUT
    //DDSNAP   DD SYSOUT=&OUT
    //PROCLIB  DD DSN=&PROCLIB,DISP=SHR
    //IEFRDER  DD DUMMY,
    //            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
    //SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
    //            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
```

```
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7JO   DD DSN=&INDEXQ..&ROOT.&FILE.PJ(0),DISP=OLD
//PAC7OJ   DD DSN=&&PAC7OJ,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAOJ,
//            DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
//PAC7PS   DD DSN=&&PAC7PS,DISP=(OLD,PASS)
//PAC7EU   DD SYSOUT=&OUT
//*
//PACA15  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACA15,PACA15$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//  COND=((00,NE,PTU004),(00,NE,CONTAJ),(00,NE,PTU400),(00,NE,PTU420))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAMM),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ$SUF DD DUMMY
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
===SEQ FOR DSM
//PAC7D3$SUFD DD DSN=&INDEXD..&ROOTD.&FILED.D3,DISP=SHR
//PAC7DC$SUFD DD DSN=&INDEXD..&ROOTD.&FILED.DC,DISP=SHR
===SEQ
//PAC7IE   DD SYSOUT=&OUT
//PAC7IF   DD SYSOUT=&OUT
//PAC7MV   DD DSN=&&PAC7OJ,DISP=(OLD,PASS)
//*
```

2.5. RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

2.5.1. RESY: INTRODUCTION

RESY: INTRODUCTION

The Database System Restoration Complement procedure (RESY) restores a Database that can be handled in on-line mode, from a System backup obtained through a utility followed by the SASY procedure.

The RESY procedure is executed after a System restoration utility to complete the restoration of the Data (AR) and Index (AN) files, and reinitializes the Journal (AJ) file.

Through the RESY procedure, the archived transactions can be recovered if 'REC' is entered on the input parameter card.

If the Journal file is not reinitialized, it must be archived prior to the System utility restoration and RESY procedures.

EXECUTION CONDITIONS

This procedure can be executed only after restoration of the AN and AR files by the on-site system utility.

On-line access must be closed.

ABEND

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

PRINTED RESULTS

The RESY procedure prints a report listing the requested options and related errors, the number of records reloaded in the database per file, the number of gaps, and the options entered in the new database.

NOTES:

1) Once the RESY procedure has been executed, the database can be used in both batch and on-line modes.

2) After the procedure execution, the current session number is the session number of the restored image, or of the most recent transaction if archived transactions were recovered.

CHECKPOINT REQUEST

This facility allows you to request synchronization points during a batch update (UPDT procedure) or during a database restoration (REST or RESY procedures).
In case of ABEND, a ROLLBACK is performed, thus securing a coherent database.

Therefore, it is always possible, after an abnormal ending of the UPDT procedure, to restart the procedure without executing a restoration. However, it is recommended to delete transactions already taken into account.

Checkpoints are performed at a frequency rate defined by the user.

EXAMPLE: A '0100' frequency rate means that a checkpoint is performed every 100 transactions.

INPUT OF THE CHECKPOINT FREQUENCY RATE FOR A BATCH UPDATE

The checkpoint frequency rate is entered on a single 'Y'-line located BEFORE the first '*'-line. The 'Y'-line is formatted as follows:

POSITION	LENGTH	VALUE	MEANING
2	1	Y	LINE CODE
4	4	nnnn	CHECKPOINT FREQUENCY RATE (DEFAULT VALUE=0000)

For the REST and RESY procedures, the checkpoint frequency is entered in the User Input.

2.5.2. RESY: USER INPUT - RESULTS

RESY : USER INPUT-RESULTS

USER INPUT

When there is no input, there are no changes to the characteristics of the database.

The input has the following structure:

POS.	LEN.	VALUE	MEANING
2	1	Y	Line code
3	7		Not used
8	2		Not used
10	1	F	French
		E	English
11	1	'0'	No suppression of journal
		'1'	Suppression of journal (update trans-
			actions are not journalized)
		' '	Retrieval of the last value
			NO INPUT EXCEPT FOR DOS/VSE
12	1		This field may ONLY be entered with
			DOS/VSE
		I	Default option (all hardware) (1)
		N	if CURRENT-DATE = DD/MM/YY
13	3	REC	if archived transactions are recov'd.
16	4	XXXX	4-character Database code chosen by
			the Database Manager (displayed in
			the top-right corner of all screens)
			DATABASE CODE IS REQUIRED WITH DSMS
			FUNCTION
20	3	nnn	Maximum access number: on-line search
			(lists) (default value: 300)
23	1	U	Implicit update (default option)
		N	Explicit update
24	4	nnnn	Checkpoint frequency rate (IMS,
			UNISYS, GCOS7, and GCOS8 only) if
			REC in col. 13 (default: nnnn=0000)
28	7		Ignored
35	12		PFkeys assigned functions (2)

POS.	LEN.	VALUE	MEANING
79	1		Dispatch option of backup:
		'D'	Dispatch Sequential backup of the database on two separate files.
		'N'	No Dispatch Standard backup on a single PC file.
		' '	Same as previous execution.

(1): This date does the following:

. Dates printed documentation,
. Checks against the system expiration date,
. Dates transaction for archiving.

Accidentally setting this date to 'N' may cause problems such as: dates reversed in printouts, blocking of the system with display of the message 'SYSTEM EXPIRATION DATE', impossibility to select archived transactions via the PACX procedure (EXPJ). It is important to check that this indicator is set correctly in each database.

(2): 12-position table, with each position corresponding to a standard function.

To modify the PFkey assigned to a function, the value of the new PFkey coded in base 36 is entered in the corresponding position in the table.
For example, to assign function 1 to PFkey 17, code 'H' in position 1 of the table.
No validation procedure is executed by the system. The PFkey assignment may be viewed on the corresponding sub-menu.

NOTES: Any field left blank defaults to the current option selection.

The default option for the language code is French.

The number of gaps cannot be specified by this procedure.

If you do not want the update transactions of the database to be saved on the Journal file, you can turn "journalization" off by setting this parameter to '1'. In this case, it is not possible to restore the database using the recovery of the archived transactions (REC parameter in the user input).

Thus, it is highly recommended that you set this parameter to '0' or leave it blank (which is the default option), in order to avoid restoration problems.

In case of error, invalid parameters are ignored, and the system ensures restoration using the parameter values stored in the sequential image of the database.

2.5.3. RESY: DESCRIPTION OF STEPS

RESY: DESCRIPTION OF STEPS

NOTE:

The default value of the &ARCHJRNL parameter is NULLFILE. For a standard operation of the database, this parameter must contain the DSNAME ARCHJRNL='$indun..$root.$file. of the archived transaction file, so that the archived transactions can be retrieved ('REC' on the parameter card of the procedure).

TRANSACTION RECOGNITION: PTU001

VALIDATION OF JOURNAL EXISTENCE: IDCAMS

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

    .0    :  The Journal file exists
    .OTHER:  The Journal file does not exist
```

VALIDATION OF JOURNAL CONTENTS: PTU380

```
This step is executed only if the Journal file exists.

.Permanent input files:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE
 -Journal file
  PAC7AJ  :    DSN=&INDEX..&ROOT.&FILE.AJ

.Output report:
  PAC7EU
  It is printed if the Journal file was not archived.

.Return codes:
 - 0:  The Journal file has been archived.
 - 8:  The Journal file was not archived. In this case, no
       other steps are executed.
```

DATABASE POSITIONING: PTU402

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

```
.Permanent output file:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR

.Permanent input file:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Input transaction file:
 -User transaction
  PAC7MB:  DSN=&&RESYMB

. Output file:
 -Work file (2 recs.)
  PAC7PS:  DSN=&&PAC7PS

.Output report:
 -Restoration report
  PAC7GZ
```

DATABASE AVAILABILITY - TRANSACTION RETRIEVAL: PTU420

This step is executed if the Journal file has been archived. It retrieves the appropriate transactions and executes an update on the first record of the Data file. It is REQUIRED for a coherent database.

```
.Input-output file:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR

.Permanent input files:
 -Journal to apply
  PAC7JO  :    DSN=&INDEXQ..&ROOT.&FILE.PJ(0)
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Input work file:
  PAC7PS:  DSN=&&PAC7PS

.Output file:
 -Update transactions
  PAC7OJ:  DSN=&&PAC7OJ  Length=167

.Output report:
 -Retrieval report
  PAC7EU

.Return code(s):
 - 0:  There are transactions to retrieve.
 - 4:  No transactions to retrieve
       OR erroneous user input.
```

In case of an abnormal end in this step, the database cannot be updated.

DATABASE UPDATE: PACA15

.Permanent update files:
 -Data file
  PAC7AR: DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN: DSN=&INDEX..&ROOT.&FILE.AN
 -Journal file
  PAC7AJ: DSN=&INDEX..&ROOT.&FILE.AJ

.Permanent input files:
 -Error message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE
 -DSMS file of VA Pac elements
  PAC7D3: DSN=&INDEXD..&ROOTD.&FILED.D3
  PAC7DC: DSN=&INDEXD..&ROOTD.&FILED.DC
  (DSM variant only)

.Input transaction file:
 -Update transactions
  PAC7MV: DSN=&&PAC7MV

.Output report(s):
 -Update report
  PAC7IE (Length=132)
 -Erroneous-transaction list
  PAC7IF (Length=132)
 (The list of transactions belonging to a user is preceded
 by a banner specifying the user code.)

.Return code(s):
 -0: OK, no error
 -2: Warning
 -4: Critical error

2.5.4. RESY: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       :  SYSTEM RESTORATION   (IBM)                       *
//*********************************************************************
//$RADP.RESY PROC FILE=$FILE,       PHYSICAL-DATABASE NUMBER
//           ROOT=$ROOT,            ROOT OF THE VA PAC SYSTEM
//           INDEX='$INDEX',        VSAM INDEX
===SEQ FOR DSM
//           ROOTD=$ROOTD,          DSMS SYSTEM ROOT
//           FILED=$FILED,          DSMS PHYSICAL-DATABASE NUMBERE
//           INDEXD='$INDEXD',      DSMS SYSTEM VSAM INDEX
===SEQ
//           INDEXP='$INDEXP',      NON-VSAM FILE INDEX
//           INDEXQ='$INDEXQ',      DATA GROUP FILE INDEX
//           SPAAJ='(TRK,(10,5),RLSE)',  JOURNAL SPACE
//           UWK=$UWK,              WORK UNIT
//           STEPLIB='$MODB',       LIBRARY OF BATCH LOAD-MODULES
//*:         SYSTCAT='$CATV',       VSAM SYSTEM CATALOG
//*:         VSAMCAT='$CATU',       VSAM USER CATALOG
//           OUT='$OUT',            OUTPUT CLASS
//           RESLIB='$RESLIB',      IMS RESLIB
//           PROCLIB='$PRCLIB',     IMS PROCLIB
//           DBDLIB='$DBDLIB',      LIBRARY OF DBD'S
//           PSBLIB='$PSBLIB',      LIBRARY OF PSB'S
//           BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//           CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//*--------------------------------------------------------------------*
//*    INSERT HERE THE SYSTEM RESTORATION STEP(S) OF DATABASES:        *
//*    'AN' & 'AR'.                                                    *
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&RESYMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//EXISAJ  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(LI&ROOT.&FILE.AJ),DISP=SHR
//*
//CONTAJ  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU380,PTU380$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,EXISAJ)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
```

```
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,DISP=SHR
//PAC7EU   DD SYSOUT=&OUT
//*
//PTU402  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU402,PTU402$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,CONTAJ)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7MB   DD DSN=&&RESYMB,DISP=(OLD,DELETE)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7PS   DD DSN=&&PAC7PS,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=140,BLKSIZE=6160),
//            SPACE=(TRK,(2,2),RLSE)
//PAC7GZ   DD SYSOUT=&OUT
//*
//PTU420  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU420,PTU420$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,CONTAJ)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7JO   DD DSN=&INDEXQ..&ROOT.&FILE.PJ(0),DISP=OLD
//PAC7OJ   DD DSN=&&PAC7OJ,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=167,BLKSIZE=6346),
//            SPACE=&SPAAJ
```

```
//PAC7PS   DD DSN=&&PAC7PS,DISP=(OLD,PASS)
//PAC7EU   DD SYSOUT=&OUT
//*
//PACA15  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACA15,PACA15$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=((00,NE,CONTAJ),(00,NE,PTU420))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAMM),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ$SUF DD DUMMY
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
===SEQ FOR DSM
//PAC7D3$SUFD DD DSN=&INDEXD..&ROOTD.&FILED.D3,DISP=SHR
//PAC7DC$SUFD DD DSN=&INDEXD..&ROOTD.&FILED.DC,DISP=SHR
===SEQ
//PAC7IE   DD SYSOUT=&OUT
//PAC7IF   DD SYSOUT=&OUT
//PAC7MV   DD DSN=&&PAC7OJ,DISP=(OLD,PASS)
//*
```

2.6. ARCH: JOURNAL ARCHIVAL

2.6.1. ARCH: INTRODUCTION

ARCH: INTRODUCTION

The Journal Archival procedure (ARCH) backs up the Journal file (AJ) as a sequential file (PJ), and re-initializes it both logically and physically.

Archived transactions do not override those transactions that were previously archived, but rather are added to them.

The archived-transaction file may be purged. Purged transactions may then be saved in another file (PQ).

Previously archived transactions can be purged, if requested. (However, non-archived journal transactions cannot be purged.)

EXECUTION CONDITION

On-line access must be closed down.

Batch procedure access authorization option: Global authorization level 4 is required.

ABENDS

If the abend occurs before the step that creates the Journal file, the procedure can be restarted as it is, after the problem has been resolved.

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

IMPORTANT:

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

2.6.2. ARCH: INPUT - RECOMMENDATIONS - RESULTS

ARCH: USER INPUT

Batch-procedure access authorization option: one '*' line with user code and password.

This procedure includes specific optional input for:

. Purging previously archived transactions that are considered obsolete. Purging may be requested up to the desired date or session number.

. Signalling the absence of previously archived trans- actions during input.

. Signalling the unavailability of the Data file (AR) during input.

. Requesting the re-initialization of the transaction file only.

The structure of this input is as follows:

POS.	LEN.	VALUE	MEANING
2	1	'S'	Line code
3	4	nnnn	Session number
7	8	ccyymmdd	OR date up to which the user requests deactivation
15	1	'I'	Absence of previously archived transactions
16	1	'D'	Data file (AR) unavailable
17	1	'J'	Re-initialization without backup, the transactions already archived are NOT retrieved on output.

The session number and the date are independent of each other. They are ignored if it is indicated that there are no input transactions (refer to paragraph 'RECOMMENDATIONS').

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

A request to re-initialize without archiving is necessary when the Journal file is physically deleted.

NOTE: In this case, the transactions which were already archived are not copied to the transaction output file. (If the Journal file is automatically catalogued by the operating system, the transactions already archived may be lost unless the file is uncatalogued).

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 only be executed if the Database is in a consistent state, and if the archived transaction file is correctly formatted.

When the Database needs to be restored after an abend or a system failure, information in the Specifications Dictionary is sometimes lost, making it impossible to execute the ARCH or the REST procedures. In this case, AND IN THIS CASE ONLY, columns 15 to 17 of the user input are to be used as follows:

. If the Data file (AR) is lost or has been flagged as 'inconsistent', a 'D' in column 16 means that the ARCH procedure will not take the Data file (AR) into account. However, the REST procedure must be executed afterward, since under these conditions, the ARCH procedure leaves the database in an inconsistent state.

. If the Journal file (AJ) is lost or destroyed, a 'J' must be entered in column 17. As a result, the ARCH procedure formats an empty Journal file. Then, the REST procedure may be executed.

. If the Journal Back-up file (PJ) is lost or destroyed, a 'I' must be entered in column 15. As a result, the ARCH procedure formats a new Journal Back-up file.

If one of these columns is accidentally set, and if the ARCH procedure is executed when the Database is in a consistent state, the consequences are:

. 'I' in col. 15: Previously archived transactions are lost. All transactions can be recovered by concatenating PJ(-1) and PJ(0) to obtain PJ(+1).

. 'D' in col. 16: The ARCH procedure must be re-executed BEFORE any update. If an update is subsequenly performed, the Database will be lost, and will have to be restored completely

. 'J' in col. 17: The contents of the Journal file are definitely lost. The output Journal file PJ, or PJ(+1) in the case of generation data files, is created empty.

PRINTED OUTPUT

This procedure prints a report stating the number of archived transactions and, if applicable, the number of records that have been 'purged'.

RESULTS

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

The Journal file (AJ) which displays transactions on-line is re-initialized.

It is also possible to store on another file all transactions that have been purged.

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

2.6.3. ARCH: DESCRIPTION OF STEPS

ARCH: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

ARCHIVAL OF JOURNAL FILE: PTU300

This step:

. Writes obsolete transactions to a special file, if the purge is requested in user input.

. Positions a flag in the Data file indicating the journal archive.

. Updates the file of archived transactions.

```
.Permanent input files:
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -Previously archived transactions
  PAC7JP:  DSN=&INDEXQ..&ROOT.&FILE.PJ(0)
 -Journal file to reinitialize
  PAC7AJ:  DSN=&INDEX..&ROOT.&FILE.AJ

.Input work file:
 -User transaction
  PAC7MB:  DSN=&&ARCHMB

.Permanent input-Output file:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR

.Output files:
 -Archived update transactions
  PAC7PJ:  DSN=&INDEXQ..&ROOT.&FILE.PJ(+1)
 -Deactivated transactions
  PAC7PQ:  DUMMY (Length=167)
  The DSN must be entered in order to keep these deactivated
  transactions.
```

.Output reports:
 -Archival report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

.Return codes:
 - 0: No error detected on the files,
 - 4: Error in journal record
      (Date or session number not numeric)
 - 8: No access authorization for batch procedure,
      OR: invalid database; in this case, restart
          the procedure with '1' in column 14
          of the user input (MBARCH).
 -12: Input-output error on a file.

CREATION OF THE JOURNAL FILE: IDCAMS

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

.Defined file:
 -Journal file
  PAC7AJ: DSN=&INDEX..&ROOT.&FILE.AJ

RE-INITIALIZATION OF THE JOURNAL FILE: PTU320

This step executes the following:

    .Creates the first record in the Journal file,

    .Re-initializes the Data file flag with the Journal
     file's address.

.Input work file:
 -User transaction
  PAC7MB: DSN=&&ARCHMB

.Permanent input/output file:
 -Data file
  PAC7AR: DSN=&INDEX..&ROOT.&FILE.AR

.Permanent input file:
 -Error message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

```
.Output file:
 -Journal file to re-initialize
  PAC7AJ:  DSN=&INDEX..&ROOT.&FILE.AJ

.Output report:
 -Review of reinitialization
  PAC7EU

.Return codes:
 - 0:  No error detected,
 - 8:  The database is invalid.
```

If the ARCH and SAVE procedures are grouped into one job, this return code can be tested in order to condition the execution of the SAVE procedure.

BACKUP CANCELLATION IN CASE OF INCONSISTENCY: IEFBR14

```
.Input file:
 -Archived update transactions
  DDPJ:  DSN=&INDEXQ..&ROOT.&FILE.PJ(+1)
```

This step is executed if the return code of the archiving program (PTU300) is other than zero.

2.6.4. ARCH: EXECUTION JCL

```
//*********************************************************************
//*  VA PAC       : TRANSACTION ARCHIVAL                              *
//*********************************************************************
//$RADP.ARCH PROC FILE=$FILE,     NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',       VSAM INDEX
//*:        SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:        VSAMCAT='$CATU',      VSAM USER CATALOG
//          OUT='$OUT',           OUTPUT CLASS
//          VOLS='$VOLO',         VOLUME OF ARCHIVED TRANSACTIONS
//          VOLU='$VOLU',         VOLUME OF DATABASE 'AJ'
//          INDEXP='$INDEXP',     INDEX OF NON VSAM FILES
//          INDEXQ='$INDEXQ',     INDEX OF DATA GROUP FILES
//          SPAPJ='(TRK,(020,10),RLSE)', SPACE FOR TRANSACTION FILE
//          SPAAJ='(CYL,(3,2),RLSE)',    SPACE FOR 'AJ' (OSAM DB)
//          UNITS='$UNITO',       TRANSACTION FILE UNIT
//          UNITU='$UNITU',       UNIT OF DATABASE 'AJ'
//          STEPLIB='$MODB',      LIBRARY OF BATCH LOAD MODULES
//          PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//          RESLIB='$RESLIB',     IMS RESLIB
//          PROCLIB='$PRCLIB',    IMS PROCLIB
//          UWK=$UWK,             WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//* INPUT       : COMMAND TO DEACTIVATE ARCHIVED TRANSACTIONS         *
//* COL  2      : 'S'                                                 *
//* COL  3 -  6 : SESSION NUMBER                                      *
//* COL  7 - 14 : DATE (CCYYMMDD)                                     *
//* COL  15     : ' '  ----> PRESENCE OF ARCHIVED TRANSACTIONS FILE   *
//*             : 'I'  ----> ABSENCE  OF ARCHIVED TRANSACTIONS FILE   *
//* COL  16     : ' '  ----> PRESENCE OF VA PAC DATABASE (PR)         *
//*             : 'D'  ----> ABSENCE  OF VA PAC DATABASE (PR)         *
//* COL  17     : ' '  ----> ARCHIVING + REINITIALIZATION             *
//*             : 'J'  ----> REINITIALIZATION WITHOUT ARCHIVING       *
//*                                                                   *
//* IF THERE IS NO INPUT (OR IF THERE IS ANY ERROR IN IT), NO DEAC-   *
//* TIVATION IS DONE, ARCHIVAL AND REINITIALIZATION ARE RUN NORMALLY  *
//*                                                                   *
//* TRANSACTIONS WITH A SESSION (DATE) LESS OR EQUAL TO THE SESSION   *
//* (DATE) INDICATED ARE NOT ARCHIVED. THEY ARE SAVED IN THE FILE     *
//* CONTAINING THE DEACTIVATED TRANSACTIONS.                          *
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&ARCHMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//PTU300  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU300,PTU300$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
```

```
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,DISP=SHR
//PAC7JP   DD DSN=&INDEXQ..&ROOT.&FILE.PJ(0),DISP=OLD
//PAC7PJ   DD DSN=&INDEXQ..&ROOT.&FILE.PJ(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=SER=&VOLS,
//            SPACE=&SPAPJ,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PJ
//PAC7MB   DD DSN=&&ARCHMB,DISP=(OLD,PASS)
//PAC7PQ   DD DUMMY,DCB=BLKSIZE=167
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//*
//OSAM1   EXEC PGM=IDCAMS,COND=(00,NE,PTU300)
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DL&ROOT.&FILE.AJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAM2   EXEC PGM=IEFBR14,COND=(00,NE,PTU300)
//PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,UNIT=&UNITU,
//            DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
//            DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//            SPACE=&SPAAJ
//*
//PTU320  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU320,PTU320$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTU300)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,DISP=SHR
//PAC7MB   DD DSN=&&ARCHMB,DISP=(OLD,DELETE)
//PAC7EU   DD SYSOUT=&OUT
//*
//DELPJ  EXEC PGM=IEFBR14,COND=(8,NE,PTU300)
//*****************************************
//DDPJ     DD DSN=&INDEXQ..&ROOT.&FILE.PJ(+1),DISP=(OLD,DELETE)
```

2.7. REOR: DATABASE REORGANIZATION

2.7.1. REOR: INTRODUCTION

REOR: INTRODUCTION

The Database Reorganization procedure (REOR) optimizes Database accesses by accounting for each deletion, and sorting the data again according to the most frequent access order.

It uses a Database backup file, PC (or 2 files when the Dispatch option is used), to rebuild one (or 2) sequential image(s). This resulting image file must then be restored via the REST procedure described above.

The functional purpose of this procedure is to rebuild the different indexes associated with all data using the 'image' of each data element. It makes the best of the system performance features since it separates historical (frozen) sessions from the current session and sorts the data in the order of the most frequent access. This makes it possible to achieve a significant reduction of the number of indexes and data items.

The REOR procedure may be used in two cases:

. When part of the data was deleted because of a malfunction or system failure, and no other procedure can be used (in particular, deletion of the AN Index file),

. When the database is to be purged of the following:

- Obsolete libraries and/or sessions;

- Entities not used in the database;

When a library is deleted, this procedure produces the same results as the Database Management (MLIB) procedure, except that it additionally deletes 'gaps'.

This procedure should be executed only on an exceptional basis, because of the special conditions concerning its use and its lengthy execution time.

Deletions taken into account by the reorganization may have been made logically by the Database update, or generated by one or several utilities. For example:

. Deletion of unused Production sessions (PEI Function)

. Deletion of entities not associated to a specific use, determined by the unused-entity extraction utility, EXPU. (See the PACX procedure in the Manual 'Batch Procedures : User's Guide'.)

EXECUTION CONDITION

If the database is available, it may remain open during reorganization since the procedure operates on sequential images of the database.

Updates executed after the back-up file used for reorganization has been built will be retrievable while the reorganized database is being restored.

Batch procedure access authorization option: Global authorization level 4 is required.

ABENDS

Refer to Chapter 'OVERVIEW', Subchapter 'Abnormal endings'.

As specified in paragraph IMPORTANT RECOMMENDATIONS below, the Reorganization procedure can be very long. It is therefore advisable to keep all temporary files after each step.

If one of the steps abends, the procedure can be restarted at the step level, but not at the procedure level.

2.7.2. REOR: INPUT - RECOMMENDATIONS

REOR: USER INPUT

Batch procedure access authorization option: one '*' line with user code and password.

Specific user input for the procedure (optional), specifying

-libraries to be purged,
-sessions to be purged or to be kept,
-entities to be purged.
-a printed copy of the list of index of the REOR procedure.

POS.	LEN.	VALUE	MEANING
2	1	'B'	Library purge
3		bbb	Library code(s): * 23
			up to 23 library codes per line

Maximum number of libraries to be purged.....: 300

POS.	LEN.	VALUE	MEANING
2	1	'V'	Purge frozen sessions
		'S'	Save frozen sessions
			Type 'V' and 'S' lines are not com-
			patible
3		ssss	Session number(s): * 17
			up to 17 session numbers per line

Maximum number of sessions indicated on the request..: 999
Maximum number of frozen sessions in a database .....: 7,500

POS.	LEN.	VALUE	MEANING
2	1	'E'	Physical purge of entities (transactions provided by EXPU)
3			Entity Type:
	1		.Type
	2	_	.UEO call code (if Type "$")
6	6	_____	Code of the entity to be purged (may be a joker code)
12	3	__	Library code
			5 groups of type/code entity/lib. possible per 'E'-type line

A Maximum number of 2,500 occurrences of an entity type is processed by the execution of the REOR procedure.

The 'List of 'purged' entities' signals when this limit is exceeded.

In case of a generic request, the entity code must be completed with *'s to make up for six characters. If the code contains six '*', all of the entity's occurrences will be deleted.

Pos.	Lon.	Valeur	Signification
2	1	'D'	PRINTED COPY OF THE LIST OF INDEX OF THE REOR PROCEDURE
3	1	' '	no report of copies of index
		'1'	report of copies of index

When the system finds an input error, it generates an error message and the procedure is not executed.

ESTIMATING FILE SIZE

The maximum sizes used during this procedure are based on the sizes of the files in the database before reorganization. The report printed by the preceding SAVE procedure provides all the relevant data:

NI = number of index file records,
ND = number of data file records MINUS number of gaps,
NC = number of primary records on the data file,
NH = number of 'frozen' (historical account) records from the data file (NH = ND - NC)

These symbols are also detailed in the presentation of each of the files for this procedure.

PRINTED OUTPUT

This procedure prints a report listing errors encountered during reorganization, and statistics on the contents of the database.

It also prints reports with the statement "IBM INTERNAL REPORT" reserving their use to IBM in case of problems.

RESULTS

The output of this procedure is a reorganized sequential image of the database (where purges may have been performed). It does not contain gaps. Gaps can be added by the REST procedure.

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

IMPORTANT RECOMMENDATIONS

The Reorganization procedure (REOR) presents a certain number of idiosyncracies of which the user should be aware:
The step that rebuilds the Index file (PTU220) uses a large amount of CPU time (around 90 per cent). If the database contains a large amount of data, it is recommended to catalog the temporary files, or to use tape files to obtain the checkpoints in case of an abend in one of the steps.

If files are transferred onto tape it is preferable to check on the initial blocking factors.

The space allocated to the sortworks should also be calculated with care.

2.7.3. REOR: DESCRIPTION OF STEPS

REOR: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VALIDATION OF USER INPUT: PTU2CL

This step validates user input and displays a return code if there is an error.

```
.Permanent input files:
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input work file:
  PAC7MB:  DSN=&&REORMB

.Output file:
 -Formatted records
  PAC7BM:  DSN=&&REORBM

.Output reports:
 -Control report
  PAC7EE
 -Batch-procedure authorization option
  PAC7DD

.Return code(s):
 - 0:  OK
 - 4:  Error on user inputs
 - 8:  No authorization for Batch procedure.
```

RETRIEVAL OF DATA: PTU200

This step selects 'data' type information in the initial sequential file of the database (in case the Dispatch option is used, it leads to the recognition of one file, that which contains the data, i.e. PC(0)). It then formats the key of each record selected for the subsequent sort.

```
.Permanent input files:
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -Sequential image of the database
  PAC7PC:  DSN=&INDEXQ..&ROOT.&FILE.PC(0)

.Output file:
 -Formatted records
  PAC7PR:  DSN=&&QS  Length=165 (Size = ND)

.Output reports:
 -Retrieval statistics
  PAC7EE
```

SORT OF DATA: SORT

```
.Input file:
 -Formatted records
  SORTIN:  DSN=&&QS
```

.Output file:
 -Sorted records
  SORTOUT: DSN=&&SQ

.Sorting criteria: SRTREO1 member of the SY PDS.

.Sort files:
  SORTWK01
  SORTWK02
  SORTWK03

EXTRACTION FOR PURGE OF ENTITIES: PTU208

This step extracts and formats the entities to be purged and indicated in the user input.

.Internal sort files:
  SORTWK01
  SORTWK02
  SORTWK03

.Input work file:
 -User transactions
  PAC7MB: DSN=&&REORBM

.Permanent input file:
 -Error messages
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

.Output file:
 -Entity records to purge
  PAC7PU: DSN=&&PU Length=13

.Output report:
 -Entity-purge transactions

PURGE: PTU210

This step purges all libraries and sessions entered in the user input. When there is no input, it formats the records.

.Internal sort
  SORTWK01
  SORTWK02
  SORTWK03

.Input work files:
 -Sorted records
  PAC7PR: DSN=&&SQ
 -Entity records to be purged
  PAC7PU: DSN=&&PU
 -User transactions
  PAC7MB: DSN=&&REORBM

.Permanent input file:
 -Error message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

.Output work files:
 -Purged records
  PAC7QS: DSN=&&QS Length=165 (Size = ND)
 -Macro-Structure call lines
  PAC7UM: DSN=&&UM Length=165

.Output reports:
 -Library and session purge report
  PAC7EE

-Entity-purge report
PAC7EK
-Technical report
PAC7EB

.Return codes:
- 0: OK
- 8: Overload of capacity

The steps that follow are executed only if the return code for the purge step is zero.

INDEX RECONSTRUCTION: PTU220

This step executes two types of procedures:

.Reconstruction of the indexes using the data

.Separation of current and frozen sessions

.Input work files:
-Purged data
PAC7UR: DSN=&&QS
-Macro-Structure call lines
PAC7UM: DSN=&&UM

.Permanent input file:
-Error message file
PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

.Output files:
-Data from frozen sessions
PAC7PA: DSN=&&PA Length=149 (Size = NH)
-Data from the current session
PAC7PB: DSN=&&PB Length=149 (Size = NC)
-First data record
PAC7PC: DSN=&&PC Length=149 (1 record)
-Temporary index file
PAC7AN: DSN=&&AN Length=55 (Size = NI)

.Work file (output, then input)
-Macro-Structure call lines
PAC7MR

.Output report:
-Index-building report
PAC7EE

SORT OF INDEXES: SORT

.Input file:
-Temporary index
SORTIN: DSN=&&AN

.Output file:
-Sorted index
SORTOUT: DSN=&&NA

.Sort criteria: SRTREO2 member of the SY PDS.

.Sort files:
SORTWK01
SORTWK02
SORTWK03

MERGE: PTU240

This step reconstructs the final sequential image using
the temporary files produced by the previous step.

```
.Permanent input file:
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input work files:
 -User transactions
  PAC7MB:  DSN=&&REORMB
 -Data from the frozen session
  PAC7PA:  DSN=&&PA
 -Data from the current session
  PAC7PB:  DSN=&&PB
 -First data record
  PAC7PC:  DSN=&&PC
 -Sorted index file
  PAC7AN:  DSN=&&NA

.Permanent output file:
 -Sequential image of the database
  PAC7CP  :    DSN=&INDEXQ..&ROOT.&FILE.PC(+1)
 If Dispatch option of backup:
 -Sequential image of the database ¾2
  PAC7PD  :    DSN=&INDEXQ..&ROOT.&FILE.PD(+1)

.Output report:
 -Logical database building
  PAC7IE
```

2.7.4. REOR: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : REORGANIZATION OF THE DATABASE                     *
//*********************************************************************
//$RADP.REOR PROC FILE=$FILE,     NO. OF THE PHYSICAL BASE
//         ROOT=$ROOT,            ROOT OF THE VA PAC SYSTEM
//         INDEX='$INDEX',        VSAM INDEX
//         INDEXP='$INDEXP',      INDEX OF NON VSAM FILES
//         INDEXQ='$INDEXQ',      INDEX OF DATA GROUP FILES
//         OUT='$OUT',            OUTPUT CLASS
//         CYL=5,                 SORT WORKS SIZE
//         SPAPC='(TRK,(100,10),RLSE)',  SPACE OF TEMPORARY FILES
//         STEPLIB='$MODB',       LIBRARY OF BATCH LOAD-MODULES
//         SORTLIB='$BIBT',       SORT LIBRARY
//         UWK=$UWK,              WORK FILE UNIT
//         VOLS='SER=$VOLO',      BACKUP FILE VOLUME
//         UNITS=$UNITO,          BACKUP FILE UNIT
//*:       SYSTCAT='$CATV',       VSAM SYSTEM CATALOG
//         PSBLIB='$PSBLIB',      LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',      LIBRARY OF DBD'S
//         RESLIB='$RESLIB',      IMS RESLIB
//         PROCLIB='$PRCLIB',     IMS PROCLIB
//        BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//        CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&REORMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//PTU2CL  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU2CL,PTU2CL$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7EE   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
```

```
//PAC7MB   DD DSN=&&REORMB,DISP=(OLD,PASS)
//PAC7BM   DD DSN=&&REORBM,DISP=(NEW,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//PTU200  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU200,PTU200$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTU2CL)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7EE   DD SYSOUT=&OUT
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(0),DISP=SHR
//PAC7PR   DD DSN=&&QS,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=165,BLKSIZE=27225)
//*
//SORTQS  EXEC PGM=SORT,COND=(00,NE,PTU200)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN   DD DSN=&&QS,DISP=(OLD,DELETE)
//SORTOUT  DD DSN=&&SQ,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=165,BLKSIZE=27225)
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(SRTREO1),DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//*
//PTU208  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU208,PTU208$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=((00,NE,PTU2CL),(00,NE,PTU200),(0,NE,SORTQS))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
```

```
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7MB   DD DSN=&&REORBM,DISP=(OLD,PASS)
//PAC7PU   DD DSN=&&PU,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=13,BLKSIZE=5200),
//            SPACE=(TRK,(1,1),RLSE)
//PAC7EE   DD SYSOUT=&OUT
//*
//PTU210  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU210,PTU210$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=((00,NE,PTU2CL),(00,NE,PTU200),(0,NE,SORTQS))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7MB   DD DSN=&&REORBM,DISP=(OLD,PASS)
//PAC7PR   DD DSN=&&SQ,DISP=(OLD,DELETE)
//PAC7PU   DD DSN=&&PU,DISP=(OLD,DELETE)
//PAC7QS   DD DSN=&&QS,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=165,BLKSIZE=27225)
//PAC7UM   DD DSN=&&UM,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=165,BLKSIZE=27225)
//PAC7EB   DD SYSOUT=&OUT
//PAC7EE   DD SYSOUT=&OUT
//PAC7EK   DD SYSOUT=&OUT
//*
//PTU220  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU220,PTU220$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=((00,NE,PTU2CL),(00,NE,PTU200),(0,NE,SORTQS),
//        (00,NE,PTU210))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
```

```
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7EE   DD SYSOUT=&OUT
//PAC7UM   DD DSN=&&UM,DISP=(OLD,PASS)
//PAC7UR   DD DSN=&&QS,DISP=(OLD,PASS)
//PAC7AN   DD DSN=&&AN,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=55,BLKSIZE=27225)
//PAC7MR   DD UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=140,BLKSIZE=6160)
//PAC7PA   DD DSN=&&PA,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=149,BLKSIZE=27267)
//PAC7PB   DD DSN=&&PB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=149,BLKSIZE=27267)
//PAC7PC   DD DSN=&&PC,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,1,RLSE),
//            DCB=(RECFM=FB,LRECL=149,BLKSIZE=149)
//*
//SORTAN  EXEC PGM=SORT,
//   COND=((00,NE,PTU2CL),(0,NE,PTU200),(00,NE,SORTQS),
//         (00,NE,PTU210))
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN   DD DSN=&&AN,DISP=(OLD,PASS)
//SORTOUT  DD DSN=&&NA,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAPC,
//            DCB=(RECFM=FB,LRECL=55,BLKSIZE=27225)
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(SRTREO2),DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//*
//PTU240  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU240,PTU240$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//   COND=((00,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
//         (0,NE,PTU210),(0,NE,SORTAN))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7IE   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&REORBM,DISP=(OLD,DELETE)
//PAC7AN   DD DSN=&&NA,DISP=(OLD,DELETE)
//PAC7PA   DD DSN=&&PA,DISP=(OLD,DELETE)
//PAC7PB   DD DSN=&&PB,DISP=(OLD,DELETE)
//PAC7PC   DD DSN=&&PC,DISP=(OLD,DELETE)
//PAC7CP   DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
```

```
//PAC7PD   DD DSN=&INDEXQ..&ROOT.&FILE.PD(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//*
```

2.8. SVAG: GENERATION-PRINT REQUEST BACKUP

2.8.1. SVAG: INTRODUCTION

SVAG: INTRODUCTION

The Generation-Print Request Backup procedure (SVAG) creates a sequential version of the file that contains the Generation-Printing Requests (AG).

The Backup file (PG) obtained is the exact image of the AG file.

EXECUTION CONDITION

The database must be closed to on-line use, in order to ensure its consistency during the backup.

Batch procedure access authorization option: global authorization level required is 4.

ABEND

The most common cause of abends is a failure to close the file to on-line access.

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

USER INPUT

Batch-procedure access authorization option: One '*' line with user code and password.

2.8.2. SVAG: DESCRIPTION OF STEPS

SVAG: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

BACKUP OF GENERATION-PRINTING REQUESTS: PTU550

```
.Input files:
 -Requests
  PAC7AG:  DSN=&INDEX..&ROOT.&FILE.AG
 -Error messages
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -User input
  PAC7MB:  DSN=&&SVAGMB

.Output file:
 -Sequential image of requests
  PAC7PG:  DSN=&INDEXQ..&ROOT.&FILE.PG(+1)

.Output reports:
 -Backup report
  PAC7EE
 -Check on procedure-access authorization
  PAC7DD

.Return code:
 - 8:  No access authorization on batch procedure.
```

BACKUP CANCELLATION IN CASE OF INCONSISTENCY: IEFBR14

```
.Input file:
 -Requests backup
  DDPG:  DSN=&INDUN..&ROOT.&FILE.PG(+1)

This step is executed if the return code of the previous
program is other than 0.
```

2.8.3. SVAG: EXECUTION JCL

```
//*********************************************************************
//*  VA PAC       : BACKUP OF 'AG' DATABASE                           *
//*********************************************************************
//$RADP.SVAG PROC FILE=$FILE,   NO. OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,         ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',     VSAM INDEX
//          INDEXP='$INDEXP',   NON VSAM FILE INDEX
//          INDEXQ='$INDEXQ',   DATA GROUP FILE INDEX
//*:        VSAMCAT='$CATU',    VSAM USER CATALOG
//*:        SYSTCAT='$CATV',    VSAM SYSTEM CATALOG
//          OUT='$OUT',         OUTPUT CLASS
//          UNITS='$UNITO',     UNIT OF BACKUP
//          VOLS='SER=$VOLO',   VOLUME OF BACKUP
//          SPAPG='(TRK,(20,5),RLSE)', SPACE OF BACKUP
//          UWK=$UWK,           WORK UNIT
//          STEPLIB='$MODB',    LIBRARY OF BATCH LOAD-MODULES
//          PSBLIB='$PSBLIB',   LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',   LIBRARY OF DBD'S
//          RESLIB='$RESLIB',   IMS RESLIB
//          PROCLIB='$PRCLIB',  IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&SVAGMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(3,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAG     DD DSN=&INDEX..&ROOT.&FILE.AG,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAG),DISP=SHR
//*
//PTU550  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU550,PTU550$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AG$SUF DD DSN=&INDEX..&ROOT.&FILE.AG,DISP=SHR
//PAC7MB   DD DSN=&&SVAGMB,DISP=(OLD,PASS)
//PAC7PG   DD DSN=&INDEXQ..&ROOT.&FILE.PG(+1),DISP=(,CATLG,DELETE),
```

```
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPAPG,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PG
//PAC7DD   DD SYSOUT=&OUT
//PAC7EE   DD SYSOUT=&OUT
//*
//DELPG  EXEC PGM=IEFBR14,COND=(8,NE,PTU550)
//*****************************************
//DDPG     DD DSN=&INDEXQ..&ROOT.&FILE.PG(+1),DISP=(OLD,DELETE)
```

2.9. REAG: GENERATION-PRINT REQUEST RESTORATION

2.9.1. REAG: INTRODUCTION

REAG: INTRODUCTION

The Generation-Print Request Restoration procedure (REAG) initializes the file containing the Generation-Printing Requests (AG), and restores or reorganizes it using the Backup file (PG) produced by the SVAG procedure.

EXECUTION CONDITION

On-line access must be closed.

Batch-procedure access authorization option:
Global authorization level required is 4.

2.9.2. REAG: USER INPUT

REAG: USER INPUT

Batch procedure access authorization option: One '*' line with user code and password.

The procedure requires the following specific input (optional):

One line to specify the request:

POS.	LEN.	VALUE	MEANING
2	2	'AG'	Line code
4	1	' '	Restoration and/or reorganization
		'I'	Initialization

One line per purge (in case of reorganization):

POS.	LEN.	VALUE	MEANING	
2	2	'AB'	Purge library commands	
		'AS'	Purge session commands	
		'AU'	Purge user commands	
4	3	bbb	Library code to be purged	('AB')
	4	ssss	Session number to be purged	('AS')
	8	uuuuuuuu	User to be purged	('AU')

Maximum number of sessions.....: 500

Maximum number of libraries.....: 100

Maximum number of users.....: 100

Default option: restoration.

2.9.3. REAG: DESCRIPTION OF STEPS

REAG: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

```
.Input file:
  CARTE

.Output file:
  PAC7MB:  DSN=&&MB

.Permanent input file:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Output report:
 -Batch-procedure authorization option:
  PAC7DD

.Return code(s):
 - 8:  No batch-procedure authorization.
```

DEFINITION OF REQUEST FILE (AG): IDCAMS

```
.Defined file:
 -Generation-printing request file:
   DSN=&INDEX..&ROOT.&FILE.AG
```

INITIALIZATION-REORGANIZATION OF REQUEST FILE (AG): PTU560

```
.Permanent input files:
 -Sequential image of requests
  PAC7PG:  DSN=&INDEXQ..&ROOT.&FILE.PG(0)
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Permanent output file:
 -Request file
  PAC7AG:  DSN=&INDEX..&ROOT.&FILE.AG

.Input transaction file:
 -User transactions
  PAC7MB:  DSN=&&REAGMB


.Output reports:
 -Restoration report
  PAC7EK
 -List of transactions
  PAC7EE
 -Batch-procedure authorization option
  PAC7DD
```

2.9.4. REAG: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac  : INITIALIZATION-RESTORATION OF THE AG FILE              *
//*********************************************************************
//$RADP.REAG PROC FILE=$FILE,  NO. OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,        ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',    VSAM INDEX
//          INDEXP='$INDEXP',  NON-VSAM FILE INDEX
//          INDEXQ='$INDEXQ',  DATA-GROUP FILE INDEX
//*:        VSAMCAT='$CATU',   VSAM USER CATALOG
//*:        SYSTCAT='$CATV',   VSAM SYSTEM CATALOG
//          OUT='$OUT',        OUTPUT CLASS
//          UWK='$UWK',        UNIT OF WORK FILE
//          STEPLIB='$MODB',   LIBRARY OF BATCH LOAD-MODULES
//          PSBLIB='$PSBLIB',  LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',  LIBRARY OF DBD'S
//          RESLIB='$RESLIB',  IMS RESLIB
//          PROCLIB='$PRCLIB', IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//* REST. OR INIT. REQUEST (DEFAULT OPTION = REST.)                   *
//* COL 02-03 : 'AG'                                                  *
//* COL 04    : ' ' FOR RESTORATION                                   *
//*           : 'I' FOR INITIALIZATION                                *
//* DELETION REQUEST (N OPTIONAL LINES)                               *
//* COL 02-06 : 'ABXXX'    DELETION OF 'XXX' LIBRARY'S COMMANDS       *
//* COL 02-07 : 'ASXXXX'   DELETION OF 'XXX' SESSION'S COMMANDS       *
//* COL 02-09 : 'AUXXXXXX' DELETION OF 'XXX' USER'S COMMANDS          *
//*********************************************************************
//*
//PTU004  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU004,PTU004$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//PAC7DD   DD SYSOUT=&OUT
//*
//DEFINE  EXEC PGM=IDCAMS,COND=(00,NE,PTU004)
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.AG),DISP=SHR
//*
//PTU560  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU560,PTU560$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
```

```
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTU004)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AG$SUF DD DSN=&INDEX..&ROOT.&FILE.AG,DISP=SHR
//PAC7MB   DD DSN=&&MB,DISP=(OLD,DELETE)
//PAC7PG   DD DSN=&INDEXQ..&ROOT.&FILE.PG(0),DISP=OLD
//PAC7DD   DD SYSOUT=&OUT
//PAC7EE   DD SYSOUT=&OUT
//PAC7EK   DD SYSOUT=&OUT
//*
```

2.10. PARM: UPDATE OF USER PARAMETERS

2.10.1. PARM: INTRODUCTION

PARM : INTRODUCTION

The User-Parameter Update procedure (PARM) updates the AE and AP User Parameter files. These files contain data that is external to the System, but which is required for its operation, i.e.:

. User codes and access authorizations,
. Codes and labels of Text entity types,
. Modifications of fixed parts of standard error messages,
. Control cards required for generation,
. System specific access key, DSMS database control (except for IBM MVS),
. Code of Security System in use (with the Security Systems Interface, in IBM MVS only), batch procedure access authorization option, blank password authorization option,
. Correspondence table for special characters.
. Association of a VisualAge Pacbase database code with a DSMS database code (IBM MVS only),
. Specific choices for the methodologies implemented in the WorkStation.

These user parameters may be updated in the following ways:

. In on-line mode, via a specific transaction (see the 'VisualAge Pacbase Interface Users'Guide').
. In batch mode, via the PARM procedure.

The PARM procedure carries out the complete user parameters management (update, print, save and restore).

NOTES:

Some user parameters must be accessible on-line:

- User codes,
- Text types (when modified by the user),
- System access keys, DSMS control,
- System security code, blank password authorization,
- System security code,
- Special characters.
- Association of a VisualAge Pacbase database code with a DSMS database code,
- WorkStation methodology choices.

These parameters are managed by the error message and on-line help documentation file (AE).

The other user parameters are only used in Batch mode by the system. They are:

- Control cards for the generated job stream,
- Modification of fixed parts of the error messages,
- Batch procedure authorization option.

The first two are managed by the AP user parameter file, and the third one by the Error message file (AE).

EXECUTION CONDITION

AE and AP files must be closed to on-line access.

ABENDS

Refer to Chapter 'OVERVIEW', Subchapter 'ABNORMAL ENDINGS'.

After correction of the problem, the procedure can be re- started as it is (provided that the User Parameters files are valid. See paragraph 'IMPORTANT RECOMMENDATION' below).

2.10.2. PARM: INPUT - RECOMMENDATIONS

PARM: USER INPUT

One line "*" (required):

POS.	LEN.	VALUE	MEANING
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

There are two types of user input control lines:

1. FILE MANAGEMENT REQUESTS:

   Backup-reloading or restoration-reloading.

2. USER PARAMETER UPDATES:

   - User codes, text types, modification of error messages, control cards;
   - System access keys;
   - DSMS control;
   - Security parameters;
   - Special characters;
   - Methodology choices.

1. FILE MANAGEMENT REQUESTS

POS.	LEN.	VALUE	MEANING
1	1		Not used
2	6	NRCHAR	BACKUP - RELOADING -Ignores the backup of input parameters (old PE) -Backs up AE and AP parameters (new PE) -Reloads AE and AP by merging the parameter backup (new PE) with AEO NOTE: This command may be performed during AE and AP updates.
2	6	NRREST	RESTORATION - RELOADING -Ignores AE and AP files -Copies the parameters of the backup in input (old PE) on the backup in output (new PE) -Reloads AE and AP by merging the parameter backup (new PE) with AEO NOTE: This command cannot be performed during AE and AP updates.

In the absence of a NRCHAR or NRREST command, the PARM procedure performs:

- The direct backup of AE and AP in the case of update transactions in input,
- The backup of AE and AP user parameters in output (new PE).

There is no AE and AP reloading. Thus, AEO cannot be taken into account.

IMPORTANT RECOMMENDATION

User parameters may be updated on-line via the User Parameter management transaction (by the updating parameters transactions or by the VisualAge Pacbase transaction for updating user codes passwords).

For this reason, the NRREST command, which does not retrieve the parameters of the AE and AP on-line files but those backed up in PE, must only be used in the following two cases:

. When AE and/or AP cannot be used; the procedure reloads AE and AP with PE and AE0, which means parameters entered on-line after the last backup are lost;

. When the characteristics of the AE and/or AP files are modified (new release of the system), the previous files can no longer be accessed by the new release: the procedure loads the new AE and AP files with PE and AE0.

These two cases REQUIRE THE USE OF THE '********' USER CODE.

See the description of procedure LOAE, used when the AE or AP files are physically lost.

2. USER PARAMETERS

2.1 User codes, text types, modification of error messages, control cards:

POS.	LEN.	VALUE	MEANING
1	1		Action code
		'C'	Creation
		'M'	Modification
		'D'	Deletion
		'B'	Multiple deletion of NC and NU lines
		' '	Creation or modification
		'X'	Creation/modification if the line contains an '&'
2	2		Line code
		'NU'	User code: Definitions and authorizations
		'NT'	Text types and names
		'NE'	Standard error message update
		'NC'	Optional control cards for generated stream
4	...	.....	Please refer to the corresponding sub-chapters for each user input

2.2 VisualAge Pacbase access keys, and DSMS database control (except IBM MVS):

POS.	LEN.	VALUE	MEANING
1	1		Action code
		'C'	Creation
		'M'	Modification
2	2	'NK'	Line code
4	3	'nnn'	Line number
7	60	.....	System access key (line '000')
			With line number = 000:
67	4	'YES '	Activation of the DSMS database control (except for IBM MVS)
		' '	No DSMS control

2.3 Security parameters: Security System Interface
(SEC extension), and two options.

POS.	LEN.	VALUE	MEANING
1	1		ACTION CODE
		'C'	CREATION
		'M'	MODIFICATION
		'D'	DELETION
2	2	'NS'	LINE CODE
4	1		SECURITY SYSTEM
		' '	NO CHANGE IN VALUE
		'&'	BLANK (DEACTIVATION)
		'R'	RACF
		'S'	TOPSECRET
5	4	cccc	RESOURCE CLASS DECLARED TO THE SECURITY SYSTEM IN RELATION TO VA PAC AUTHORIZATIONS.
9	1		VA PAC RESOURCE DEFINITION FOR EACH USER:
		' ' or '&'	DEFINITION MUST BE DONE IN THE SECURITY SYSTEM TABLES.
		'P'	DEFINITION MUST BE DONE IN VA PAC (BATCH: NU LINES; ON-LINE: PU CHOICE)
			RACF ONLY
10	1	' ' or '&'	POSSIBILITY OF ENTERING A USER CODE - PASSWORD DIFFERENT FROM THAT OF THE INITIAL SCREEN CONNECTION AND '*' LINES
		'N'	NO POSSIBILITY OF ENTERING ANOTHER USER CODE - PASSWORD.
11	1		BATCH PROCEDURE ACCESS AUTHORIZATION:
		' '	NO CHANGE IN VALUE
		'0'	NO AUTHORIZATION VALIDATION (DEFAULT VALUE FOR CREATION)
		'1'	AUTHORIZATION VALIDATION
12	1		BLANK PASSWORD AUTHORIZATION OPTION:
		' '	NO CHANGE IN VALUE
		'0'	AUTHORIZATION OF BLANK PASSWORDS (DEFAULT VALUE FOR CREATION)
		'1'	BLANK PASSWORDS NOT AUTHORIZED

NOTE: When a security system is operating on the database user codes (input code 'NU', on-line choice 'PU') are ignored. For more details, refer to the SECURITY SYSTEMS INTERFACE Reference Manual.

2.4 Correspondence table for special characters of keywords

Keywords for entity names are converted into upper-case letters, but accented letters are not, making keyword searches complicated. In order to convert these special characters, add a line NW. For example, to convert é ----> E

POS.	LEN.	VALUE	MEANING
1	1		Action code
		'C'	Creation
		'M'	Modification
		'A'	Deletion
2	2	'NW'	Line code
4	1	é	Initial character
5	1	E	Converted character
6	1	E	Associated uppercase

2.5 Association of VisualAge Pacbase database codes to DSMS database codes (IBM MVS only)

POS.	LEN.	VALUE	MEANING
1	1		Action code
		'C'	Creation
		'M'	Modification
		'A'	Deletion
2	2	'NB'	Line code
4	4	pppp	Logical VisualAge Pacbase database name
8	4	dddd	DSMS database code

2.6 Definition of methodology choices for the WorkStation

The transactions with which these lines must be defined (NL and NM codes) are supplied with the installation deck. Refer to the 'ENVIRONMENT & INSTALLATION' Manual, Chapter 'INSTALLATION', Subchapter 'DATABASE COMPLEMENT: WORKSTATION INSTALLATION' for more details on the loading of these transactions.

2.10.3. PARM: USER-CODE DEFINITION

DEFINITION OF USER CODES

System user codes are stored in the Error Message file. To update user codes, you have to fill in batch form 'NU', which is described below.

Each user is identified by a code and a password which are entered in order to access the Database (whether in batch or on-line), the User Parameter Management transaction, and the Production Environment Interface (PEI) function.

Each user is assigned access rights, or AUTHORIZATIONS. These rights are organized according to the following hierarchy:

1. GLOBAL AUTHORIZATION LEVEL

- Access to a network's libraries (all databases)
- Access to the management of user parameters
- Access to batch procedures

2. AUTHORIZATION LEVEL ASSOCIATED TO A VA PAC DATABASE

- Access to the database's libraries (all libraries)
- Access to the database's batch procedures
- Access to the database's PEI Environment Function

2. AUTHORIZATION LEVEL ASSOCIATED TO A DATABASE LIBRARY

When a lower authorization level is entered, it has precedence over the higher level.

LIBRARY ACCESS AUTHORIZATIONS

The authorization levels are:

. Access prohibited
. Read only
. Current session update
. All-session update

The global authorization allows access to the entire database BUT the libraries explicitly mentionned.

If the GLOBAL and PER DATABASE authorization levels are not specified (access prohibited), the user is authorized to access only those libraries that are explicitly mentioned.

NOTES:

The character '&' sets the global or per database authorization level to blank.

It is recommended to grant the lowest global authorization, since it is both easier and safer to codify authorized libraries than prohibited ones.

Example:

To grant a read-only authorization on all libraries except the 'AP1' library, on which updates will be authorized, specify:

. '1' in the GLOBAL AUTHORIZATION level or the DATABASE AUTHORIZATION level,

. '3' in the LIBRARY AUTHORIZATION specific to 'AP1'.

Access authorization in the Inter-Library (***) mode may also be granted.

Update of a library-authorization level

The update of library-specific authorizations is performed on a terminal/work station basis. Modification of an authorization should be performed on the work station for which it was granted.

In order to cancel access to a library, just enter zero as its authorization level.

Access authorization in the Inter-Library (***) mode may also be granted.

NOTES

No check is performed on library codes. If a library is mentioned several times with different authorization levels, only the first occurrence will be taken into account.

No consistency check is performed between the global authorization and the specific authorizations. For a given level of global authorization, the same level may be given for one or several libraries within the same database.

USER-PARAMETER MANAGEMENT ACCESS AUTHORIZATION

The authorization levels are:

```
0     :  Access prohibited
1     :  Read-only access
2 or 3:  Update access
4     :  Administrator only
```

(See the explanation below.)

BATCH PROCEDURE ACCESS AUTHORIZATION (option)

If the option of batch-procedure authorization check is active (see paragraph '2. User Parameters' above) the user will be able to run the batch procedures according to the authorization level granted.

Refer also to the paragraph mentioning this option in Chapter 'OVERVIEW', Subchapter 'Access Rights', where a table lists the authorizations required for each procedure.

PEI FUNCTION ACCESS AUTHORIZATION

Three authorization levels are associated to the Production Environment Interface (PEI) Function:

```
0      :  Access prohibited
1      :  Read-only access
2, 3, 4:  Update access
```

A PEI authorization is entered like a special library codes, '$E', in an authorization area specific to a library.

2.10.4. PARM: USER-CODE GLOBAL AUTHORIZATIONS

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	8		USER CODE (REQ. IN CREATION) Each user must be given a personal user code and associated password. For each user code, the system defines the libraries which can be accessed and the actions allowed (read, update of current session, update of all sessions). The user code is stored for each transaction in the Journal. The management of user codes and access authorizations is the responsibility of the Database Administrator, who can be consulted for information on each user's access authorizations.
2	3	NUMER. 000	LINE NUMBER (REQ. IN CREATION) General definition line of a user (code, password and global authorization). Used as the key.
3	8		USER PASSWORD The password is associated with a user code. Using blanks between two characters is forbidden. NOTE: On sites using the Security Systems Interface (RACF or TOPSECRET), passwords are managed by the Security System, not by the VA-Pac user code management function.
4	1	 Blank 0 1 2 3	GENERAL AUTHORIZATION LEVEL This authorization grants access to the Database. No global access authorization. No global access authorization. Read-only access authorized for both current and all frozen sessions. Read-write access authorized for the current session and read-only access for all frozen sessions. Read-write access is authorized for both current and test sessions. NOTE: This authorization is limited by the provisions of the PROTECTION OF EXTRACTED ENTITIES and MODIFICATION OF EXTRACTED LINES fields on the Library Definition screen of the libraries concerned.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		4	Update is authorized on any session. The provisions of the PROTECTION OF EXTRACTED ENTITIES and MODIFICATION OF EXTRACTED LINES fields on the Library Definition screens are NOT taken into account. Moreover, the administrator has the right to initialize libraries, unlock locked entities, and update frozen-session labels.
5	1	NUMER.	USER-PARAMETER UPDATE AUTHORIZATION This level concerns authorizations for the user-parameter management access.
		Blank	Access prohibited.
		0	Access prohibited.
		1	Read-only access.
		2 or 3	Read-write access.
		4	Administrator's authorization.
6	1		GENERAL AUTHORIZATION ON PROCEDURES
		Blank	No authorization on the batch procedures.
		0	No authorization on the batch procedures (default option in creation)
		2	AUTHORIZATION ON STANDARD EXTRACTIONS Level allowing access to standard extractors.
		3	AUTHORIZATION ON SPECIAL EXTRACTIONS "Project Manager" level: Level granting access to special procedures.
		4	MAXIMUM AUTHORIZATION "VisualAge Pacbase Manager" level: Access to the database management, generation-print and PEI file management procedures. NOTE: This level can be granted for a global authorization only.
7	30		USER NAME Name may be entered in lower-case print.
8	15		COMMENTS ON USER This may be entered in lower-case print.

2.10.5. PARM: USER-CODE SPECIFIC AUTHORIZATIONS

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	8		USER CODE Each user must be given a personal user code and associated password. For each user code, the system defines the libraries which can be accessed and the actions allowed (read, update of current session, update of all sessions). The user code is stored for each transaction in the Journal. The management of user codes and access authorizations is the responsibility of the Database Administrator, who can be consulted for information on each user's access authorizations.
2	3		LINE NUMBER It is advisable to leave gaps in the line numbering sequence in order to facilitate future insertions.
		1 to 999	SPECIFIC AUTHORIZATION: - on libraries, - on the PEI function.
3	4		DATABASE CODE FOR MULTI-DATABASE SITES ONLY. Logical name of the database. This code is displayed in the identifier which appears in the top right corner of all screens. It is used to establish the relation between a VA-Pacbase database and a DSMS database. No validity check is performed here.
			LIBRARY ACCESS TABLE NOMBRE DE REPETITIONS : 15 Two access types may be entered: - Access to a Database library, - Access to the Production Environment Interface (PEI function).
	3		LIBRARY CODE
		BBB	Code identifying the selected library.
		***	Read-only access authorization on the whole database

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			('Inter-library' mode).
		$E	Access to Production Environment Interface function.
	1		SPECIFIC AUTHORIZATION LEVEL
		0	Access not authorized. However, you can view, from a lower Library, the entities defined in this Library.
		1	Consultation of all sessions.
		2	Consultation of all sessions and update of the current session.
		3	Consultation and update of all sessions. NOTE: This authorization is limited by the provisions of the PROTECTION OF EXTRACTED ENTITIES and MODIFICATION OF EXTRACTED LINES fields (library definition).
		4	Consultation and update of all sessions, authorization to perform 'database' management operations but only within the Library specified in the preceding field. NOTE: The provisions of the PROTECTION OF EXTRACTED ENTITIES and MODIFICATION OF EXTRACTED LINES fields (Library Definition) are NOT taken into account.
			ACCESS TO PEI FUNCTION ($E):
		1	Consultation only.
		2 3 or 4	Consultation and update.
6	1		DATABASE AUTHORIZATION LEVEL
		Blank	No authorization on the database.
		0	No authorization on the database.
		1	Read-only on current session, Read-only on archived sessions.
		2	Read-write on current session, Read-only on archived sessions.
		3	Read-write on current session, Read-write on archived sessions.
		4	All authorizations.
7	1		BATCH PROCEDURE AUTHORIZATION LEVEL

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		Blank	No authorization on the batch procedures.
		0	No authorization on the batch procedures.
		2	AUTHORIZATION ON STANDARD EXTRACTIONS on the database.
		3	AUTHORIZATION ON SPECIAL EXTRACTIONS on the database.

2.10.6. PARM: TEXT TYPES

PARM: TEXT TYPES

UPDATING TEXT TYPES

Each text entity is defined in the database by a definition line (batch) or definition screen (on-line). They both include a TYPE OF TEXT field. (For more details, refer to the SPECIFICATIONS DICTIONARY Reference Manual).

All sets of TYPE OF TEXT and NAME OF TEXT TYPE are stored in the Error Message file and can be updated via Batch Form 'NT'.

Updating includes creation, modification or deletion in the file.

NOTE: When a text type is deleted, the corresponding label becomes 'UNKNOWN TYPE'.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1		LANGUAGE INDICATOR
		F	French.
		E	English.
2	2		TYPE OF TEXT (REQ. IN CREATION) The TYPE OF TEXT field is used for documentation purposes only, and allows the user to: .obtain the list of texts sorted by type (CHOICE: LTT), .have explicit titles including the labels corresponding to the chosen type of text, on screens and reports which contain the text. The coding of types and labels depends on an external parameter handled by the Database Administrator.
		T	Default value.
3	15		NAME OF TEXT TYPE (REQ. IN CREATION) Specify the label to appear with the corresponding Type of Text. NOTE: This label will appear on the Text Definition screen when the corresponding Type of Text is used, and on screens and reports which contain the text. Enter the name to appear with the corresponding Type of Text. This name will appear on the Text Definition screen when the corresponding Type of Text is used.

2.10.7. PARM: MODIFICATION OF STANDARD ERROR MESSAGES

MODIFICATIONS OF STANDARD ERROR MESSAGES

The first part of standard error messages for applications generated by the system may be modified if the default options are not suitable.

The second part of a standard error message cannot be modified since it is the data element's clear name.

Updating is performed by filling in Batch Form 'NE', which is described below.

NOTES

Modifications cannot be made on error messages specific to the System. Only error messages related to a given application can be modified.

Default options are taken into account after the deletion of a record in the User Parameter file (AP).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1		LANGUAGE INDICATOR
		F	French.
		E	English.
2	2		ERROR CODE (REQ. IN CREATION) This is the code that the user must enter to modify the first part of the standard error message.
		2	To modify 'INVALID ABSENCE FOR THE FIELD'
		3	To modify 'INVALID PRESENCE FOR THE FIELD'
		4A	To modify 'NON-ALPHABETICAL CLASS FIELD'
		4Z	To modify 'NON-NUMERICAL CLASS FIELD'
		5	To modify 'INVALID VALUE FOR THE FIELD'
		8F	To modify 'INVALID CREATION RECORD'
		9F	To modify 'INVALID DELETE/MODIFY RECORD'
		9G	To modify 'END OF LIST'
			PACBENCH C/S ERROR MESSAGES
		DUPL	To modify 'INVALID CREATION RECORD'
		NFND	To modify 'INVALID DELETE/MODIFY RECORD'
		END	To modify 'END OF LIST'
		ABSC	To modify 'ABSENCE OF RECORD'
3	30		FIRST PART OF ERROR MESSAGE (REQ. IN CREATION) Enter the message to appear before the erroneous data element name for the corresponding Error Code. Note: This message will be stored in the User Parameter file (AP).

2.10.8. PARM: GENERATED-STREAM CONTROL CARDS

PARM: GENERATED-STREAM CONTROL CARDS

Generated job streams of batch or on-line programs, or database descriptions, must include the job control commands necessary for subsequent processing, such as program assembly, compilation or link-edit.

NOTE: A job stream is made up of several programs of a given type (batch or on-line program, screen, or database description). It is generated by the system for a specific user during a given session and originates from a particular library.

These job control commands have a two-fold purpose:

. They are used to separate two programs, screens or database descriptions,
. They control the execution of necessary procedures in the job stream.

Job control commands can be located at different points in the job stream:

. At the beginning of the generated job stream,
. Just before a program, screen or database description,
. Immediately following a program, screen, or database description,
. At the end of the generated job stream.

Each job control command is made up of one or several control cards, identified by an option code. Each card is made up of a line of Job Control Language. This JCL can be in packed format, allowing certain variable data to be parameterized (such as program code, screen code, library code).

This information is stored in the User Parameter file (AP). Some standard options are supplied with the system. database administrator may modify these options or create new ones.

Optional control card updating is accomplished via Batch Form 'NC'.

In addition, the screen obtained with 'LNC' in the choice field displays the list of the various control cards sets.

CALL OF CONTROL CARDS

When a user requests the generation of a program, screen or database description, he/she must call the set of control cards necessary to process the job stream. They are identified by their OPTION CODE and are found in the User Parameter file.

The user must do the following:

. Enter the job-stream 'front/back' option codes on the Library Definition screen,

. Enter the program 'front/back' option codes on the Library Definition screen (they will be the default options for all programs in that library),

. Enter the program 'front/back' options on the Program Definition screen if the default options are not appropriate,

. Enter on-line program- and map- 'front/back' options on the Screen Definition screen,

. Enter data-block 'front/back' options on the Database Block Definition screen.

The Generation and Print Commands (GP) screen may be used to modify the options specified at the library-, program-, or screen-level. The modified options will be taken into account for the current run only.

The priority order of requests for one run of the generation process is the following: generation request, then Entity definition file, then library.

Job stream cards are called by a special command, FLx, where 'x' is the type of generated Entity.

PARAMETERIZATION OF CONTROL CARDS

Job control cards are parameterized according to the following principles:

A control card consists of three types of information:

. A fixed part, representing the syntax of the job control language in use,

. A first variable part, made up of components that can be determined in advance (such as the generated program code or the library name),

. A second variable part, made up of fields that can be entered only at the last minute, because they depend on the run to be executed. (For example, SYSOUT class and time limit.)

The two variable parts of a control card are supplied by the decoding of the value in the INSERTION REFERENCE CHARACTER field. This character will replace the variable parts in the control card image entered in the file.

It is specified in the line's last character.

Five parameters are available for a line. The five positions preceding the Insertion Reference character contain their symbolic values.

When the control cards are generated, the INSERTION REFERENCE CHARACTER is decoded and the system replaces it with the corresponding parameter values according to the following rules:

. Alphabetic parameters whose values are given in the input descriptions will be decoded in terms of their pre-established meaning.

. Numeric parameters introduced on the screen or in the generation-print request transaction are decoded in terms of their user-specified meaning.

EXAMPLE

Suppose a user wants to insert the following control card before all generated programs:

```
**COMPIL  DATE:MM/DD/YY,PROG:PPPPPP,TIME:D,CLASS:C
```

Let '-' be the INSERTION REFERENCE CHARACTER defined by the user; the card will have the following pattern:

```
**COMPIL  DATE:-,PROG:-,TIME:-,CLASS:-,
```

The parameters to be entered should be in the order 'DP12', where:

.'D'= Date, determined by the system.

.'P'= Generated program code.

.'1'= The number '1' parameter entered by the user on the Generation and Print Commands (GP) screen in the format '1=D', either at the job stream level (FLP) if it is a default option, or else at the program level (GP).

.'2'= Replacement parameter number '2' in the format '2=C', entered in the same way as parameter '1' above.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1		TYPE OF OPTION (REQ. IN CREATION)
		A	Beginning of generated program job stream.
		D	Before the generated program.
		F	Following the generated program.
		Z	Following the generated program job stream.
2	1		OPTION CODE (REQ. IN CREATION) Identifies optional job control cards. To be specified for: - The 'Front/Back' of the job stream on the Library Definition screen, - The 'Front/Back' program options on the Library Definition screen or the Program Definition screen, - The 'Front/Back' options for the on-line program and for the map on the Screen Definition screen, - The 'Front/Back' block options on the Block Definition screen.
3	2		LINE NUMBER (REQ. IN CREATION)
		BLANK	Option title line:
		0 - 99	Title in the "Optional Card Image" field. Lower-case keying accepted.
		NUMERICAL	Optional control card: It is recommended to leave gaps in a line's number sequence in order to make future insertions possible.
4	67		OPTIONAL CONTROL CARD IMAGE The image of the optional control card is written in compressed format. Parameterized information is represented by the INSERTION REFERENCE CHARACTER(S). The last column of this field (67th) is specified with the label "C". Any value other than blank entered in this column will be generated in column 72 of the control card. This field accepts lowercase characters.
			INPUT PARAMETERS Each of these parameters selects a data element from the internal or source system library:
		A	Library code ('*' entity, 1 to 3 characters).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		B	Source library name ('*' entity, 1-36 characters).
		C	Current date including century (10 characters).
		D	Current date determined by the system, in eight-character format.
		G	Session number of the database when the job runs (5 characters).
		I	DSMS change number
		J	Name of the job initialized by the System (IMS only).
		K	No. of the job initialized by the System (IMS only).
		L	Parameter required for operation of the VA Pac-Endevor Interface. It may also be used to suit user needs. Its purpose is to select the data provided by Pacbase Constants, in the following format: EEntityNomexterBasBibSessTjj/mm/aahh:mm:ssUsercode With: E (1) = Entity type (O, M for Map, P, or B) Entity (6) = VisualAge Pacbase Entity code Nomexter (8) = External name Base (4) = Database code Bib (3) = Library code Sess (4) = Generation session number T (1) = Session status (T or blank) dd/mm/yy (8) = Generation date or mm/dd/yy, according to the format used in the documentation. hh/mm/ss (8) = Generation time Usercode (8) = User code for generation
		N	Sequence number of program in the generated program job stream (2 characters).
		P	External name of generated program, screen or block.
		Q	Class code of generated program (Batch language generator). Dialog code (dialog generator or Pacbench C/S)
		R	Clear name of generated program, screen, or block (from definition screen).
		S	Code of generated program, screen or block.
		U	User code.
		V	Job stream number (two-digit value), automatically assigned according to the order of execution.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		1 to 9	Numerical values of input parameters will be decoded according to the values on the GENERATION AND PRINT COMMANDS (GP) screen. NOTE: This field accepts lowercase characters.
5	1		INPUT PARAMETER NO.1 Can take any one of the values as defined above. Can take on any of the values defined above as well as numerical values.
6	1		INPUT PARAMETER NO.2 Can take any one of the values defined above. Can take on any of the values defined above as well as numerical values.
7	1		INPUT PARAMETER NO.3 Can take any one of the values defined above. Can take on any of the values defined above as well as numerical values.
8	1		INPUT PARAMETER NO.4 Can take any one of the values defined above. Can take on any of the values defined above as well as numerical values.
9	1		INPUT PARAMETER NO.5 Can take any one of the values defined above. Can take on any of the values defined above as well as numerical values.
10	1		INSERTION REFERENCE CHARACTER This is a given character that will be replaced, in the generated control card, by the values of the input parameter codes. The first occurrence of this character is replaced by the field selected by the first non-blank input parameter. Only the first non-blank characters of the field are taken into account. When the first character in the field is blank, insertion reference is suppressed. This is a given character that will be replaced, in the generated control card, by the values of the input parameter codes. The first occurrence of this character is replaced by the field selected by the first non-blank input para-

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			meter. Only the first non-blank characters of the field are taken into account. When the first character in the field is blank, insertion reference is removed (except for parameters B and R). The second occurrence of this character is replaced by the field selected by the second non-blank input parameter. This continues through the last occurrence, until the end of the Optional Control Card Image, or until the length of the line is 71 characters. Insertion Reference Characters which have not been replaced, as well as those which correspond to an erroneous input parameter, will remain unchanged.

2.10.9. PARM: DESCRIPTION OF STEPS

PARM: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

UPDATE AND BACKUP: PACU15

This step executes the direct update of parameters in the Error Message (AE) and User Parameters (AP) files.

It automatically backs-up the parameters in PE(+1).

WARNING: If NRREST is requested, the backup PE(+1) is the image of PE(0), which is the previous backup, and not the backup of the AE and AP files.

```
.Permanent input-output files:
 -Error messages
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -User parameters
  PAC7AP:  DSN=&INDEX..&ROOT.&ROOT.AP

.Permanent input files:
 -User parameter backup
  PAC7EC:  DSN=&INDEXQ..&ROOT.&ROOT.PE(0)

.Transaction file:
 -Update transactions
  PAC7MC:    DSN=&&PARMMB

.Output file
 -User parameter backup
  PAC7CE:  DSN=&INDEXQ..&ROOT.&ROOT.PE(+1)

.Output reports
 -Printing of the update file and review
  PAC7IJ
 -Check on procedure access authorization
  PAC7DD
```

```
.Return codes:
  0:  OK - Reloading of the AE and AP files
  4:  OK - No reloading of the AE and AP files.
  8:  No parameter-update authorization
```

REDEFINITION OF AE AND AP FILES: IDCAMS

This step is executed only if the reloading or restoration of the AE and AP files was requested.
It executes a DELETE/DEFINE on the AE and AP files.

RECONSTRUCTION OF THE AE AND AP FILES: PACU80

This step is executed only if the reloading or restoration of the AE and AP files was requested.

```
.Permanent input files:
 -User parameter backup
  PAC7CE:  DSN=&INDEXQ..&ROOT.&ROOT.PE(+1)
 -Initial sequential image of
  error messages
  PAC7LE:  DSN=&INDEXP..&ROOT.&ROOT.AE0

.Transaction file:
 -Update transactions
  PAC7MC:    DSN=&&PARMMB

.Permanent output files:
 -Error messages to be rebuilt
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -User parameters to be recreated
  PAC7AP:  DSN=&INDEX..&ROOT.&ROOT.AP

.Output report:
 -Reconstruction report
  PAC7IJ

.Sort file(s):
  SORTWK01
  SORTWK02
  SORTWK03
```

DELETE OF USER PARAMETER BACKUP FILE: IEFBR14

This step is executed only if the parameter-update authorization has been denied.

```
. Input file:
 - User-parameter backup
   DDPE:  DSN=&INDSN..&ROOT.&ROOT.PE(+1)
```

2.10.10. PARM: EXECUTION JCL

```
//*********************************************************************
//*  VisualAge Pacbase : UPDATE OF LOCAL PARAMETERS
//*********************************************************************
//$RADP.PARM PROC ROOT=$ROOT,   ROOT OF THE VA PAC SYSTEM
//           INDEX='$INDEX',    VSAM INDEX
//*:         SYSTCAT='$CATV',   VSAM SYSTEM CATALOG
//           OUT='$OUT',        OUTPUT CLASS
//           INDEXP='$INDEXP',  INDEX OF NON-VSAM FILES
//           INDEXQ='$INDEXQ',  INDEX OF DATA GROUP FILES
//           SPAPE='(TRK,(10,1),RLSE)', BACKUP SPACE
//           UNITS='$UNITO',    BACKUP UNIT
//           VOLS='SER=$VOLO',  PARAMETERS BACKUP VOLUME
//           STEPLIB='$MODB',   LIBRARY OF BATCH LOAD-MODULES
//           SORTLIB='$BIBT',   LIBRARY OF SORT
//           PSBLIB='$PSBLIB',  LIBRARY OF PSB'S
//           DBDLIB='$DBDLIB',  LIBRARY OF DBD'S
//           RESLIB='$RESLIB',  IMS RESLIB
//           PROCLIB='$PRCLIB', IMS PROCLIB
//           UWK=$UWK,          WORK UNIT
//           CYL=1,             SIZE OF SORTWORK
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PARMMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAP     DD DSN=&INDEX..&ROOT.&ROOT.AP,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAP),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//PACU15  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACU15,PACU15$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AP$SUF DD DSN=&INDEX..&ROOT.&ROOT.AP,DISP=SHR
//PAC7CE   DD DSN=&INDEXQ..&ROOT.&ROOT.PE(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPE,
//            DCB=&INDEXQ..DSCB.&ROOT.&ROOT.PE
//PAC7DD   DD SYSOUT=&OUT
```

```
//PAC7EC   DD DSN=&INDEXQ..&ROOT.&ROOT.PE(0),DISP=SHR
//PAC7IJ   DD SYSOUT=&OUT
//PAC7MC   DD DSN=&&PARMMB,DISP=(OLD,PASS)
//*
//DEFINE  EXEC PGM=IDCAMS,COND=(00,NE,PACU15)
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AP),DISP=SHR
//*
//PACU80  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACU80,PACU80$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PACU15)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD UNIT=&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)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AP$SUF DD DSN=&INDEX..&ROOT.&ROOT.AP,DISP=SHR
//PAC7CE   DD DSN=&INDEXQ..&ROOT.&ROOT.PE(+1),DISP=(OLD,KEEP)
//PAC7IJ   DD SYSOUT=&OUT
//PAC7LE   DD DSN=&INDEXP..&ROOT.&ROOT.AE0,DISP=SHR
//PAC7MC   DD DSN=&&PARMMB,DISP=(OLD,DELETE)
//*
//DELPE  EXEC PGM=IEFBR14,COND=(8,NE,PACU15)
//*****************************************
//DDPE     DD DSN=&INDEXQ..&ROOT.&ROOT.PE(+1),DISP=(OLD,DELETE)
```

3. VERSIONING UTILITIES

3.1. PEI: PRODUCTION ENVIRONMENT INTERFACE

3.1.1. PEI: OVERVIEW

PEI: INTRODUCTION

The Production Environment Interface is an optional facility, and its use depends upon the corresponding purchase agreement.

The purpose of the Production Environment Interface facility is to provide:

. Management of all GENERATION ENVIRONMENTS defined on-site (production, system acceptance, test, etc.);

. Follow-up of entities generated from the database and managed in any on-site environment;

. Automatic session freeze when needed (for example, when generating into a production environment);

. The possibility to manually request a session freeze;

. Generation of purge requests for redundant frozen sessions;

. A list of frozen sessions for which there were entities put into production;

. Information related to these entities, such as the library code, the code of the user, and the session number of the last generation and of the most recent database freeze;

. Project(s) follow-up by development team(s) in relation to generated entities.

For further information, refer to the PRODUCTION ENVIRONMENT INTERFACE Reference Manual.

PEI FILES

The management of environments and that of entities in production use the same logical file.

In order for this file to be updatable simultaneously in on-line and batch modes, it is physically duplicated in two 'mirror' files, one being dedicated to on-line update, the other to batch update.

For read-only accesses, the system uses the most recent update of the file.

FILE SIZE

These two files may be accessed directly or sequentially depending on which type of processing is to be performed.

Length: 110 bytes, key (length: 26, position 1)

```
N = number of records
E = number of production environments
G = average number of generated entities per library
L = number of loadlibs where a given entity is used
B = number of libraries in the database
S = number of production sessions
```

$N = E + (G * B * L * 2) + S$

L must be equal to at least 2, since a given entity may be used both in a development and a production environment.

Each deletion is logical until a restoration procedure is performed.

Both files (on-line and batch) should be the same size.

3.1.2. INPE: FILE INITIALIZATION
3.1.2.1. INPE: INTRODUCTION

INPE: INTRODUCTION

The PEI File Initialization procedure (INPE) initializes the PEI file backup. This procedure must be run whenever the Database is initialized or a previous release is retrieved.

Its execution precedes the Restoration procedure (RSPE) in order to initialize the PEI files (AB and AC).

EXECUTION CONDITION

The AB and AC files must be closed to on-line use. The database files may stay open.

Batch procedure access authorization option: Authorization level 4 is required.

ABENDS

Once the problem has been solved, the INPE procedure may be restarted as it is.

USER INPUT

Batch procedure access authorization option: One '*' line with user code and password.

3.1.2.2. INPE: DESCRIPTION OF STEPS

INPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

PEI INITIAL BACKUP: PACR01

```
.Permanent input files:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input file:
 -User input file
  PAC7MB:    DSN=&&INPEMB

.Output file:
 -PEI initial backup
  PAC7PP:  DSN=&INDEXQ..&ROOT.&FILE.PP(+1)

.Output reports:
 -Execution report
  PAC7IB
 -Batch-procedure authorization option
  PAC7DD

.Sort file(s):
  SORTWK01
  SORTWK02

.Return code(s):
 8:  No batch-procedure authorization
```

3.1.2.3. INPE: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : PEI OPTION - BACKUP FILE INITIALIZATION           *
//*********************************************************************
//$RADP.INPE PROC FILE=$FILE,    NUMBER OF THE PHYSICAL DATABASE
//         ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//         INDEX='$INDEX',       INDEX OF USER VSAM FILES
//         INDEXP='$INDEXP',     INDEX OF SYSTEM NON-VSAM FILES
//         INDEXQ='$INDEXQ',     DATA GROUP FILE INDEX
//*:       SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:       VSAMCAT='$CATU',      VSAM USER CATALOG
//         STEPLIB='$MODB',      LIBRARY OF LOAD-MODULES
//         SORTLIB='$BIBT',      SORT LIBRARY
//         OUT='$OUT',           OUTPUT CLASS
//         UWK=$UWK,             WORK UNIT
//         CYL=1,                SIZE OF SORTWORK
//         UNITS='$UNITO',       BACKUP UNIT
//         VOLS='$VOLO',         BACKUP VOLUME
//         SPAPP='(TRK,(5,2),RLSE)',    BACKUP SPACE
//         PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//         RESLIB='$RESLIB',     IMS RESLIB
//         PROCLIB='$PRCLIB',    IMS PROCLIB
//        BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//        CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&INPEMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*
//VERIFY  EXEC  PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PACR01  EXEC  PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACR01,PACR01$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
```

```
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7IB   DD SYSOUT=&OUT,DCB=BLKSIZE=101
//PAC7MB   DD DSN=&&INPEMB,DISP=(OLD,PASS)
//PAC7PP   DD DSN=&INDEXQ..&ROOT.&FILE.PP(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=SER=&VOLS,
//            SPACE=&SPAPP,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PP
//*-------------------------------------------------------*
//*          EXECUTE RESTORATION PROCEDURE : 'RSPE'       *
//*-------------------------------------------------------*
//*
```

3.1.3. SVPE: FILE BACKUP
3.1.3.1. SVPE: INTRODUCTION

SVPE: INTRODUCTION

The PEI File Backup procedure (SVPE) formats the AB and AC PEI files sequentially into one file (PP).

EXECUTION CONDITION

The AB and AC files must be closed to on-line use.

Batch procedure access authorization option: Authorization level 4 is required.

ABNORMAL EXECUTION

Most abends are the result of forgetting to close the files to on-line use.

Once the problem has been solved, the SVPE procedure can be re-started as it is.

USER INPUT

Batch procedure access authorization option: One '*' line with user code and password.

3.1.3.2. SVPE: DESCRIPTION OF STEPS

SVPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

PEI BACKUP: PACR60

```
.Permanent input files:
 -'Batch' PEI file
  PAC7AB:  DSN=&INDEX..&ROOT.&FILE.AB
 -'On-line' PEI file
  PAC7AC:  DSN=&INDEX..&ROOT.&FILE.AC
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Output file:
 -PEI backup
  PAC7PP:  DSN=&INDEXQ..&ROOT.&FILE.PP(+1)

.Input file:
 -Transaction file
  PAC7MB:  DSN=&&SVPEMB

.Output reports:
 -Execution report
  PAC7IE
 -Batch-procedure authorization option
  PAC7DD

.Return code(s):
 8:  User not authorized
```

BACKUP CANCELLATION IN CASE OF INCOHERENCE: IEFBR14

```
.Input file:
 -PEI Backup
  DDPP:  DSN=&INDEXQ..&ROOT.&FILE.PP(+1)

This step is executed if the return code of the previous
program is other than 0.
```

3.1.3.3. SVPE: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : PEI OPTION - FILE BACKUP                          *
//*********************************************************************
//$RADP.SVPE PROC FILE=$FILE,     NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',       INDEX OF USER VSAM FILES
//          INDEXP='$INDEXP',     INDEX OF SYSTEM NON-VSAM FILES
//          INDEXQ='$INDEXQ',     DATA GROUP FILE INDEX
//*:        SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:        VSAMCAT='$CATU',      VSAM USER CATALOG
//          STEPLIB='$MODB',      LIBRARY OF LOAD-MODULES
//          OUT='$OUT',           OUTPUT CLASS
//          UNITS='$UNITO',       BACKUP UNIT
//          VOLS='$VOLO',         BACKUP VOLUME
//          SPAPP='(TRK,(5,2),RLSE)',    BACKUP SPACE
//          UWK=$UWK,             WORK UNIT
//          PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//          RESLIB='$RESLIB',     IMS RESLIB
//          PROCLIB='$PRCLIB',    IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&SVPEMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*
//VERIFY  EXEC  PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAB     DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//DDAC     DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//PACR60  EXEC  PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACR60,PACR60$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AB$SUF DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC$SUF DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
```

```
//PAC7DD   DD SYSOUT=&OUT
//PAC7IE   DD SYSOUT=&OUT,DCB=BLKSIZE=81
//PAC7MB   DD DSN=&&SVPEMB,DISP=(OLD,PASS)
//PAC7PP   DD DSN=&INDEXQ..&ROOT.&FILE.PP(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=SER=&VOLS,
//            SPACE=&SPAPP,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PP
//*
//DELPP  EXEC PGM=IEFBR14,COND=(8,NE,PACR60)
//*****************************************
//DDPP     DD DSN=&INDEXQ..&ROOT.&FILE.PP(+1),DISP=(OLD,DELETE)
```

3.1.4. RSPE: FILE RESTORATION
3.1.4.1. RSPE: INTRODUCTION

RSPE: PEI FILE RESTORATION

RSPE: INTRODUCTION

The RSPE procedure recreates the PEI files, AB and AC, from the sequential image obtained with the SVPE procedure.

EXECUTION CONDITION

The AB and AC files must be closed to on-line use.

Batch procedure authorization option: Authorization level 4 is required.

Since the RSPE procedure recreates the PEI files, it is advisable to have previously readjusted the file sizes according to their estimated size evolution. These modifications must be made in the System Parameters library (SY).

ABNORMAL EXECUTION

Once the problem is solved, the RSPE procedure can be restarted as it is.

USER INPUT

Batch procedure authorization option:
One '*' line with user code and password.

3.1.4.2. RSPE: DESCRIPTION OF STEPS

RSPE: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

```
.Input file:
  CARTE

.Output file:
  PAC7MB:  DSN=&&MB

.Permanent input file:
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Output report:
 -Batch-procedure authorization option:
  PAC7DD

.Return code(s):
 - 8:  No batch-procedure authorization.
```

VERIFICATION OF VSAM FILES: IDCAMS

PEI RESTORATION: PACR61

```
.Input file:
 -User input
  PAC7MB:  DSN=&&RSPEMB

.Permanent input files:
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -PEI backup file
  PAC7PP:  DSN=&INDEXQ..&ROOT.&FILE.PP(0)

.Permanent output files:
 -'Batch' PEI file
  PAC7AB:  DSN=&INDEX..&ROOT.&FILE.AB
 -'On-line' PEI file
  PAC7AC:  DSN=&INDEX..&ROOT.&FILE.AC


.Output reports:
 -Review
  PAC7IF
 -Batch-procedure authorization option
  PAC7DD

.Return code:
  8:  No authorization on batch procedures
```

3.1.4.3. RSPE: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : PEI OPTION - FILE RESTORATION                     *
//*********************************************************************
//$RADP.RSPE PROC FILE=$FILE,    NUMBER OF THE PHYSICAL DATABASE
//         ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//         INDEX='$INDEX',       INDEX OF USER VSAM FILES
//         INDEXP='$INDEXP',     INDEX OF SYSTEM NON VSAM FILES
//         INDEXQ='$INDEXQ',     DATA GROUP FILE INDEX
//*:       SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:       VSAMCAT='$CATU',      VSAM USER CATALOG
//         STEPLIB='$MODB',      LIBRARY OF LOAD-MODULES
//         UWK=$UWK,             WORK UNIT
//         OUT='$OUT',           OUTPUT CLASS
//         PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//         RESLIB='$RESLIB',     IMS RESLIB
//         PROCLIB='$PRCLIB',    IMS PROCLIB
//        BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//        CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//*
//PTU004  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU004,PTU004$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//PAC7DD   DD SYSOUT=&OUT
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAB     DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//DDAC     DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//DELDEF  EXEC  PGM=IDCAMS,COND=(00,NE,PTU004)
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.AB),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.AC),DISP=SHR
//*
//PACR61  EXEC  PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACR61,PACR61$SUG,&BUF,
```

```
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTU004)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AB$SUF DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC$SUF DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7IF   DD SYSOUT=&OUT,DCB=BLKSIZE=81
//PAC7MB   DD DSN=&&MB,DISP=(OLD,PASS)
//PAC7PP   DD DSN=&INDEXQ..&ROOT.&FILE.PP(0),DISP=SHR
//*
```

3.1.5. PRPE: PRODUCTION ENVIRONMENT PRINTOUTS
3.1.5.1. PRPE: INTRODUCTION

PRPE: INTRODUCTION

The PEI Printing procedure (PRPE) prints data related to the Production Environment Interface.

EXECUTION CONDITION

None, the files can remain open for on-line processing.

Batch-procedure authorization option: Authorization level 2 is required.

ABENDS

Once the problem is solved, the PRPE procedure can be restarted as it is.

3.1.5.2. PRPE: USER INPUT

PRPE: USER INPUT

Batch-procedure access authorization:
One '*' line with user code and password.

Specific input:

POS.	LEN.	VALUE	MEANING
2	2	'PL'	Line code
4	1	'1'	List of environments by library
5	1	'1'	List of libraries by environment
6	1	'1'	List of entities in production, by environment
7	1	'1'	List of entities in production, by session
8	1	'1'	List of environments by entity (entities sorted by VA Pac codes)
9	1	'1'	List of environments by entity (entities sorted by external names)

In order to exclude one or more of these lists, leave the corresponding position to blank.

Only the first parameter line is taken into account; any other input is ignored by the system.

3.1.5.3. PRPE: DESCRIPTION OF STEPS

PRPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

PEI PRINTING: PACR10

```
.Permanent input files:
 -'Batch' PEI file
  PAC7AB:  DSN=&INDEX..&ROOT.&FILE.AB
 -'On-line' PEI file
  PAC7AC:  DSN=&INDEX..&ROOT.&FILE.AC
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input transaction file:
 -Printing requests
  PAC7MB:    DSN=&&PRPEMB

.Output reports:
 -Printouts
  PAC7IE
 -Batch-procedure authorization option
  PAC7DD

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03

.Return code(s):
 8:  No authorization on Batch procedures
```

3.1.5.4. PRPE: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac        : PEI OPTION - PRODUCTION ENVIRONMENT PRINTING     *
//*********************************************************************
//$RADP.PRPE PROC FILE=$FILE,     NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//          INDEXP='$INDEXP',     INDEX OF NON-VSAM FILES
//          INDEX='$INDEX',       INDEX OF VSAM FILES
//          SORTLIB='$BIBT',      SORT LIBRARY
//*:        SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:        VSAMCAT='$CATU',      VSAM USER CATALOG
//          OUT='$OUT',           OUTPUT CLASS
//          CYL=1,                SIZE OF SORTWORK
//          STEPLIB='$MODB',      LIBRARY OF LOAD MODULES
//          PSBLIB='$PSBLIB',     IMS PSBLIB
//          DBDLIB='$DBDLIB',     IMS DBDLIB
//          RESLIB='$RESLIB',     IMS RESLIB
//          PROCLIB='$PRCLIB',    IMS WORKLIB
//          UWK=$UWK,             WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//*  INPUT   -   PRINTING REQUEST                                     *
//*  COL 02-03 : 'PL' (CARD CODE)                                     *
//*  COL 04    : '1'  ENVIRONMENTS PER LIBRARY                        *
//*  COL 05    : '1'  LIBRARIES PER ENVIRONMENT                       *
//*  COL 06    : '1'  ENTITIES PER ENVIRONMENT                        *
//*  COL 07    : '1'  ENTITIES PER SESSION                            *
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PRPEMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAB     DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//DDAC     DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PACR10  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACR10,PACR10$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
```

```
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AB$SUF DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC$SUF DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7IE   DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB   DD DSN=&&PRPEMB,DISP=(OLD,DELETE)
//*
```

3.1.6. GRPE: TRANSACTION-GENERATION FOR REORGANIZATION
3.1.6.1. GRPE: INTRODUCTION

GRPE: INTRODUCTION

The Transaction-Generation for Reorganization procedure (GRPE) generates deletion transactions used as input to the Database Reorganization (REOR) procedure. These transactions purge the frozen sessions of the database which are not production sessions.

PRINT

The GRPE procedure prints a comparative report on frozen sessions and production sessions.

EXECUTION CONDITION

None, the files can remain open for on-line processing.

Batch-procedure authorization option: Authorization level 4 is required.

ABENDS

Once the problem has been solved, the GRPE procedure can be restarted as it is.

USER INPUT

Batch procedure authorization option: One '*' line with user code and password.

3.1.6.2. GRPE: DESCRIPTION OF STEPS

GRPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

GENERATION OF TRANSACTIONS FOR REORGANIZATION: PACR40

```
.Permanent input files:
 -'Batch' PEI file
   PAC7AB:  DSN=&INDEX..&ROOT.&FILE.AB
 -'On-line' PEI file
   PAC7AC:  DSN=&INDEX..&ROOT.&FILE.AC
 -Data file
   PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
   PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Error message file
   PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input file:
 -User input
  PAC7MB:  DSN=&&GRPEMB

.Output file:
 -Generated trans. for reorganization
  PAC7MV:  DSN=&&GRPEREOR

.Output reports:
 -Execution report
  PAC7IK
 -Batch-procedure authorization option
  PAC7DD

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03

.Return code(s):
 8:  No authorization on Batch procedures.
```

3.1.6.3. GRPE: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac      : PEI OPTION - GENERATION OF TRANSACTIONS FOR        *
//*                REORGANIZATION OF THE VA Pac DATABASE              *
//*********************************************************************
//$RADP.GRPE PROC FILE=$FILE,  NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,        ROOT OF THE VA PAC SYSTEM
//          INDEXP='$INDEXP',  INDEX OF NON VSAM FILES
//          INDEX='$INDEX',    INDEX OF  VSAM FILES
//          SORTLIB='$BIBT',   SORT LIBRARY
//*:        VSAMCAT='$CATU',   VSAM USER CATALOG
//*:        SYSTCAT='$CATV',   VSAM SYSTEM CATALOG
//          OUT='$OUT',        OUTPUT CLASS
//          CYL=1,             SIZE OF SORTWORK
//          STEPLIB='$MODB',   LIBRARY OF LOAD MODULES
//          PSBLIB='$PSBLIB',  IMS PSBLIB
//          DBDLIB='$DBDLIB',  IMS DBDLIB
//          RESLIB='$RESLIB',  IMS RESLIB
//          PROCLIB='$PRCLIB', IMS WORKLIB
//          UWK=$UWK,          WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*-------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&GRPEMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAB     DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//DDAC     DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PACR40  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACR40,PACR40$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
```

```
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AB$SUF DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC$SUF DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7IK   DD SYSOUT=&OUT,DCB=BLKSIZE=101
//PAC7MB   DD DSN=&&GRPEMB,DISP=(OLD,PASS)
//PAC7MV   DD DSN=&&GRPEREOR,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//            SPACE=(TRK,(10,1),RLSE)
//*
```

3.1.7. HIPE: AUTOMATIC SESSION FREEZE
3.1.7.1. HIPE: INTRODUCTION

HIPE: INTRODUCTION

The Automatic Freeze Session procedure (HIPE) freezes the current session of the database when entities are put into production. It then prints a list of entities in production.

EXECUTION CONDITION

The database files and the PEI files (AB and AC) must be closed to on-line processing.

ABENDS

Once the problem is resolved, the HIPE procedure can be restarted as it is.

3.1.7.2. HIPE: USER INPUT

HIPE: USER INPUT

A required '*' line:

POS.	LEN.	VALUE	MEANING
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	'***'	Inter-library (required)

An optional session freeze line:

POS.	LEN.	VALUE	MEANING
2	2	'X1'	Line code if the entities have been put into production
		'X4'	if no entity has been put into production
4	4	'HIST'	Freeze request
8	60		Freeze comments
68	4	ssss	Forcing of session number (number comprised between current session number +1 and current session number +100)

If this line is not entered, it is automatically generated when entities are put into production.

This line may be entered in order to:

.Give a specific freeze comment,

.Force the session number.

PRINTED REPORTS

The HIPE procedure prints a report and a list of the entities used in production, if the database has been frozen.

3.1.7.3. HIPE: DESCRIPTION OF STEPS

HIPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE CONSISTENCY CHECK: PTUBAS

```
.Permanent input files:
 -Data file
  PAC7AR  :    DSN=&INDEX..&ROOT.&FILE.AR
 -Error message file
  PAC7AE  :    DSN=&INDEX..&ROOT.&ROOT.AE

.Output report
 -Validity report (Length=079)
  PAC7DS

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

AUTOMATIC SESSION FREEZE: PACR30

```
.Permanent input files:
 -'Batch' PEI file
  PAC7AB:  DSN=&INDEX..&ROOT.&FILE.AB
 -'On-line' PEI file
  PAC7AC:  DSN=&INDEX..&ROOT.&FILE.AC
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Journal file
  PAC7AJ:  DSN=&INDEX..&ROOT.&FILE.AJ
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE


.Input transaction file:
 -Session freeze requests
  PAC7MB:  DSN=&&HIPEMB

.Output report:
 -Execution report
  PAC7IG

.Work files:
  PAC7MW
  PAC7WB
```

3.1.7.4. HIPE: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : PEI OPTION - SESSION FREEZE                       *
//*********************************************************************
//$RADP.HIPE PROC FILE=$FILE,     NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//          INDEXP='$INDEXP',     INDEX OF NON-VSAM FILES
//          INDEX='$INDEX',       INDEX OF VSAM FILES
//*:        SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:        VSAMCAT='$CATU',      VSAM USER CATALOG
//          OUT='$OUT',           OUTPUT CLASS
//          STEPLIB='$MODB',      LIBRARY OF LOAD MODULES
//          PSBLIB='$PSBLIB',     IMS PSBLIB
//          DBDLIB='$DBDLIB',     IMS DBDLIB
//          RESLIB='$RESLIB',     IMS RESLIB
//          PROCLIB='$PRCLIB',    IMS WORKLIB
//          UWK=$UWK,             WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//*  INPUT    : REQUESTED '* ' LINE                                   *
//*             OPTIONAL  'X1' LINE (AUTOMATICALLY GENERATED)        *
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&HIPEMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAB     DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//DDAC     DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PTUBAS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUBAS,PTUBAS$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DS   DD SYSOUT=&OUT
//*
//PACR30  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
```

```
//        PARM=(DLI,PACR30,PACR30$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AB$SUF DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC$SUF DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7IG   DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB   DD DSN=&&HIPEMB,DISP=(OLD,DELETE)
//PAC7MW   DD UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=167,BLKSIZE=6346),
//            SPACE=(TRK,(1,1),RLSE)
//PAC7WB   DD UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=110,BLKSIZE=6160),
//            SPACE=(TRK,(2,1),RLSE)
//*
```

3.1.8. SIPE: PRODUCTION TURNOVER SIMULATION
3.1.8.1. SIPE: INTRODUCTION

SIPE: INTRODUCTION

The Production Turnover Simulation procedure (SIPE) simulates a production turnover via a batch update of the PEI files. For that purpose, it processes user input specifying the characteristics of the entities that are to be used in production.

Three SIPE operations are available:

1. Simulation of update with GPRT:

   Generated entities are entered as batch update transactions where generation data is entered.

2. Simulation of environment transfer:

   Same operation as above, except that generation data comes from the source environment.

3. Existing systems retrieval:

   Same operation as in 1. above; the procedure is executed only once after the system is initialized via the INPE procedure.

EXECUTION CONDITION

None, since the database is not directly updated. Only the AB file is updated in the same way as it is by GPRT.

Batch procedure access authorization: Level 3 is required.

ABENDS

After solving the problem, you can restart the procedure as it is.

3.1.8.2. SIPE: USER INPUT

SIPE: USER INPUT

A required '*' line.

POS.	LEN.	VALUE	MEANING
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Library code (required)
22	4	ssss	Session number (blank if current)
26	1		Session status (' ' or 'T')
59	8	CCYYMMDD	Generation date, if session is not current (input field for a frozen session of type blank or T - not an input field of current session)

One 'EE' line identifying the environment (required):

POS.	LEN.	VALUE	MEANING
2	2	'EE'	Line code
4	1	t	Entity type: 'B','M','O','P', or 'U'
5	1	r	Target environment type
6	1	s	Source environment type

One 'EU' line for each entity to update:

POS.	LEN.	VALUE	MEANING
2	2	'EU'	Line code
4	8	cccccccc	Entity code
12	8	eeeeeeee	Entity external name in target environment if different from code in Database
20	8	nnnnnnnn	Entity external name in source environment if transfer with RENAME

3.1.8.3. SIPE: DESCRIPTION OF STEPS

SIPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

PRODUCTION TURNOVER: PACR22

```
.Permanent input files:
 -'Batch' PEI file
  PAC7AB:  DSN=&INDEX..&ROOT.&FILE.AB
 -'On-line' PEI file
  PAC7AC:  DSN=&INDEX..&ROOT.&FILE.AC
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Transaction file:
 -User input
  PAC7MB:  DSN=&&SIPEMB

.Output file:
 -Transactions used to build data cards
  for TRANSFER utilities
  PAC7MT:  DSN=&&PAC7MT  Length=80

.Output reports:
 -Execution report
  PAC7IE
 -Batch-procedure authorization option
  PAC7DD
```

3.1.8.4. SIPE: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : PEI OPTION - PRODUCTION TURNOVER SIMULATION       *
//*********************************************************************
//$RADP.SIPE PROC FILE=$FILE,   PHYSICAL-DATABASE NUMBER
//           ROOT=$ROOT,        ROOT OF THE VA PAC SYSTEM
//           INDEX='$INDEX',    VSAM INDEX
//           INDEXP='$INDEXP',  INDEX OF NON-VSAM FILES
//*:         SYSTCAT='$CATV',   VSAM SYSTEM CATALOG
//*:         VSAMCAT='$CATU',   VSAM USER CATALOG
//           OUT='$OUT',        OUTPUT CLASS
//           STEPLIB='$MODB',   LIBRARY OF BATCH LOAD-MODULES
//           PSBLIB='$PSBLIB',  LIBRARY OF PSB'S
//           DBDLIB='$DBDLIB',  LIBRARY OF DBD'S
//           RESLIB='$RESLIB',  IMS RESLIB
//           PROCLIB='$PRCLIB', IMS PROCLIB
//           UWK=$UWK,          WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//* INPUT : LINE '*' : REQUIRED                                       *
//*          LINE 'EE': REQUIRED (IDENTIFICATION ENTITIES AND         *
//*                     ENVIRONMENT)                                  *
//*         LINE 'EU': ONE 'EU' LINE PER ENTITY                       *
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&SIPEMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAB     DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//DDAC     DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PACR22  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACR22,PACR22$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AB$SUF DD DSN=&INDEX..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC$SUF DD DSN=&INDEX..&ROOT.&FILE.AC,DISP=SHR
```

```
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7MB   DD DSN=&&SIPEMB,DISP=(OLD,DELETE)
//PAC7MT   DD DSN=&&PAC7MT,DISP=(,PASS),UNIT=&UWK,
//            DCB=(LRECL=80,RECFM=FB,BLKSIZE=3440),
//            SPACE=(TRK,(2,2),RLSE)
//PAC7DD   DD SYSOUT=&OUT
//PAC7IE   DD SYSOUT=&OUT
//*
```

3.2. PAC/TRANSFER

PAC/TRANSFER: INTRODUCTION

The purpose of the Pac/transfer facility is to provide an easy versioning of the developments made in a VisualAge Pacbase Database; it automates transfers of update transactions from one session to one or several sessions.

Pac/transfer scans the archived Journal file and reads a dedicated Parameter file.

One or several source environments are defined in this parameter file. Each can correspond with one or more target environments.

Pac/transfer selects, from the archived Journal file, transactions that match the criteria defined via these parameters.

Pac/transfer then generates transactions for the target environment(s) defined in the parameter file.

These transactions are used by the VA Pac batch update procedure (UPDT). If the VA Pac Database is under DSMS control, such updates are automatically included in this control.

FUNCTIONALITIES

The objective of Pac/transfer is to transfer updates made in a given session to one or several target sessions.

Once a development is completed in a test session, it is possible to transfer this session's contents onto another validation-dedicated session, and, if necessary, onto another session dedicated to production-turnover.

In the transfer file, the selected transactions from the source session are duplicated as many times as there are target sessions.

There are no constraints regarding the chronological order of sessions. It is possible to transfer a source session's status onto a later target session (target-session number greater than that of the source session), just as it is possible to transfer it onto a previous target session (target-session number lesser than that of the source session).

OPERATING MODE

1. UPDATING THE TRANSFER PARAMETERS

Process to be executed if there are new Transaction Sets to be defined, or if parameters of existing Sets are to be modified.

2. COMPRESSING THE ARCHIVED JOURNAL

Optional process (depending on the site).

3. CREATING THE TRANSFER FILE

4. PREPARING THE DSMS ENVIRONMENT

Process to be executed only if the Database is under DSMS control.

5. GENERATING THE TRANSFER TRANSACTIONS

6. UPDATING THE VISUALAGE PACBASE DATABASE

7. REINITIALIZING THE DSMS ENVIRONMENT

Process to be executed only if the Database is under DSMS control.

3.2.1. TRUP: TRANSFER-PARAMETER UPDATE
3.2.1.1. TRUP: INTRODUCTION

TRUP: INTRODUCTION

Pac/transfer's processing is based on the user-defined parameters stored in the UV parameters file. These parameters control the various processes of the facility's procedures.

These parameters must be created -- via a TRUP execution -- prior to any Pac/transfer operation. Any change to one of these parameters must be followed by a new TRUP execution.

Several sets of transfer parameters, called Transaction Sets, may be defined. The parameter file can therefore store several Transaction Sets.

By defining several Transaction Sets, you can make your transfer operations very flexible and adapt them fully to your own requirements.

Transfer parameters -- described below -- define one Transaction Set. It is not possible to set parameters common to all Sets.

TRANSFER PARAMETERS

1.1. SESSION:

Specify one source session and at least one target session.

If you specify several target sessions, transactions entered in the source session will be transferred to each specified target session.

NOTE: For each transfer request line, you must specify an order number so as to ensure the adequate chronology of transfers. This is particularly important when several source sessions have the same target session.

1.2. LIBRARY:

As a default, ALL Libraries in the VisualAge Pacbase Database are taken into account for the requested source session, and the transfer target are the same Libraries.

You may restrict the scope of a transfer by selecting one particular source Library, which then becomes the default target Library. This means that you have the wider option of selecting one or more target Libraries.

NOTE: If the source Library is to be part of the selected target Libraries, specify its code explicitely.

If you specify several target Libraries, transactions relating to the selected source Library will be transferred to each of the target Libraries.

EXAMPLE: When a transfer is defined from one source session to TWO target sessions, and from one source Library to THREE target Libraries, the volume of transferred transactions will be SIX times larger than the volume of selected transactions.

1.3. USER:

As a default, transactions entered by ANY Database user are transferred under a unique user code.

You may restrict the scope of the transfer by selecting one particular source user-code, which will be considered as the default target user-code. You may therefore also select a target user-code different from the selected source user-code.

1.4. DSMS CHANGE NUMBER:

>>>>> This type of selection refers to VisualAge Pacbase Databases under DSMS control only.

As a default, transactions associated to ANY Change are transferred under the same Change number.

You may restrict the scope of the transfer by selecting one particular source Change-number, which will be considered as the default target Change-number. You may also select a target Change-number different from the source Change-number.

It is also possible to transfer all transactions under a single target user-code.

NOTE: This option overrides any target user selection such as described in Paragraph 1.3.

EXECUTION CONDITION

None.

PRINTED REPORT

Printout of the parameter-file contents.

3.2.1.2. TRUP: USER INPUT

TRUP: USER INPUT

. User identification line (required)

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

. Session-selection line

Within a Transaction Set, there must be at least one selection line of this type.

Pos.	Len.	Val.	Meaning
1	1		Action code:
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	5	ttttt	Transaction Set code (required)
			NOTE:'99999' is not an authorized value
7	2	'GS'	Line type
9	4		Source Session (required)
18	3		Continuation line number, if you need to define more than 14 target sessions
			NOTE: All prior input in the preceding line must be repeated in the continuation line.
21	56		Target session(s)
			(at least one session is required)
			Session numbers are entered without the 'T' and are not separated by blanks
77	4		Transfer order number (required)

. Library-selection line

Pos.	Len.	Val.	Meaning
1	1		Action code:
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	5	ttttt	Transaction Set code (required)
7	2	'GB'	Line type
9	3		Source Library (required)
18	3		Continuation line number, if you need to define more than 20 target Libraries NOTE: All prior input in the preceding line must be repeated in the continuation line.
21	60		Target Library(ies) Default: source Library Library codes are not separated by blanks.

. User-selection line

Pos.	Len.	Val.	Meaning
1	1		Action code
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	5	ttttt	Transaction Set Code (required)
7	2	'GU'	Line type
9	8		Source user (required)
21	8		Target user Default: source user

. DSMS-change selection line

Pos.	Len.	Val.	Meaning
1	1		Action code:
		'C'	Creation
		'M'	Modification
		'D'	Deletion
2	5	sssss	Transaction-Set Code (required)
7	2	'GC'	Line type
9	3		Source product code (required)
			NOTE: The product code must be left-justified.
12	6		Source Change number (required)
18	3		Target selection type:
		'000'	Change selection (default)
		'001'	User selection
			NOTE: If you use both selection types all prior input in the 2nd line must be identical to that of the first line.
			.IF SELECTION TYPE = 000:
21	3		Target product code
			NOTE: The product code must be left-justified.
24	6		Target Change number
			Default: Source product/Change
			.IF SELECTION TYPE = 001:
21	8		Target user code
			Default: Source user

. Multiple-deletion request line

Multiple deletions may be requested at two levels: for the complete Transaction Set or for all selections of a given type made for the selected Set.

Pos.	Len.	Val.	Meaning
1	1	'B'	Multiple deletion request
2	5	llllll	Transaction Set Code (required)
		'GS'	Deletion of complete Set (default)
		'GB'	Deletion of Library selections
		'GU'	Deletion of user selections
		'GC'	Deletion of Change selections

EXAMPLES:

EXAMPLE 1

Transfer of transactions entered in a frozen session (3050T) to another frozen session (3000T).

```
*USER   PASSWORD
CLot1 GS3050       3000                                1
```

EXAMPLE 2

Same as above, but with an additional target session: the current session (9999).

```
*USER   PASSWORD
CLot1 GS3050      30009999                             1
```

EXAMPLE 3

Same as Example 2 plus additional source selections: Transactions must have been entered in the BIB Library, by the user JEAN, in relation to Changes 'PR 001220' and 'PR 001250'.

```
*USER PASSWORD
CLot1 GS3050       30009999                            1
CLot1 GBBIB
CLot1 GCPR 001220
CLot1 GCPR 001250
CLot1 GUJEAN
```

EXAMPLE 4

Transactions made in two different sessions must be transferred to the same target session. The sequence number (far right, in Position 77) specifies the order of transfers.

```
*USER PASSWORD
CLot1 GS3050        3000                               2
CLot1 GS4000        3000                               1
```

EXAMPLE 5

Transactions entered in session 3050T in relation to Change 'PR 001220' are transferred to session 3000T, assigned to Change 'PR 001250' under user code JEAN.

```
*USER PASSWORD
CLot1 GS3050        3000                               1
CLot1 GCPR 001220   PR 001250
CLot1 GCPR 001220001JEAN
```

3.2.1.3. TRUP: DESCRIPTION OF STEPS

TRUP: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

UPDATE OF THE SELECTION PARAMETERS: PTUG10

This step updates the selection-parameter file.

```
.Permanent input files:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Transaction file
 -User input
  PAC7MA:  DSN=&&TRUPMB

.Output file:
 -List of Transfer Sets
  PAC7ML:  DSN=&&TRUPML

.Input/output file:
 -Parameter file
  PAC7UV:  DSN=&INDEX..&ROOT.&FILE.UV

.Work file:
 -Transaction file with generated multiple deletions
  PAC7MV:  DSN=&&TRUPMV

.Output reports:
 -Input check
  PAC7ET
 -User check
  PAC7DD
```

SELECTION-PARAMETER PRINTOUT: PTUG11

```
.Permanent input files:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -Parameter file
  PAC7UV:  DSN=&INDEX..&ROOT.&FILE.UV

.Output file:
 -List of target sessions
  PAC7GL:  DSN=&&TRUPGL

.Output report:
 -Printout of parameter table
  PAC7ET
```

PRINTING OF TARGET-SESSION LIST: PTUG12

```
.Input files:
 -Data file
  PAC7AR:  DSN=&INDEX.&ROOT.&FILE.AR
 -Parameter file
  PAC7UV:  DSN=&INDEX.&ROOT.&FILE.UV
 -Error-message file
  PAC7AE:  DSN=&INDEX.&ROOT.&ROOT.AE
 -Target-session list
  PAC7GL:  DSN=&&TRUPGL
 -List of Sets
  PAC7ML:  DSN=&&TRUPML

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03

.Output report:
 -Target-session list printout
  PAC7ET
```

3.2.1.4. TRUP: EXECUTION JCL

```
//*********************************************************************
//*  Pactransfer     :    UPDATE OF USER PARAMETERS
//*  PAC/TRANSFER  :    UPDATE OF THE TRANSFER PARAMETERS             *
//*********************************************************************
//$RADP.TRUP   PROC FILE=$FILE, VA PAC PHYSICAL-DATABASE NUMBER
//         ROOT=$ROOT,          RADICAL OF THE VA PAC SYSTEM
//         INDEX='$INDEX',      INDEX OF VA PAC VSAM FILES
//         INDEXP='$INDEXP',    INDEX OF NON-VSAM FILES
//*:       VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//         OUT=$OUT,            OUTPUT CLASS
//         STEPLIB='$MODB',     LOAD-MODULE LIBRARY
//         PSBLIB='$PSBLIB',    PSB LIBRARY
//         DBDLIB='$DBDLIB',    DBD LIBRARY
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         UWK=$UWK,            WORK UNIT
//    SPAGL='(TRK,(30,10))',    SPACE OF FILE 'FH'
//    SORTLIB='$BIBT',          SORT LIBRARY
//    CYL=20,                   SIZE OF SORT-WORK FILES
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TRUPMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//PTUG10  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG10,PTUG10$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF   DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV       DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7ET   DD SYSOUT=&OUT
//PAC7MA   DD DSN=&&TRUPMB,DISP=(OLD,DELETE)
//PAC7ML   DD DSN=&&TRUPML,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7MV   DD DSN=&&TRUPMV,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//*
//PTUG11  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
```

```
//        PARM=(DLI,PTUG11,PTUG11$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(8,LE,PTUG10)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT  DD SYSOUT=&OUT
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV       DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7GL   DD DSN=&&TRUPGL,DISP=(,PASS),
//            UNIT=&UWK,SPACE=&SPAGL,
//            DCB=(RECFM=FB,LRECL=19,BLKSIZE=1900)
//PAC7ET   DD SYSOUT=&OUT
//*
//PTUG12  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG12,PTUG12$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(8,LE,PTUG10)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT  DD SYSOUT=&OUT
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV       DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7ML       DD DSN=&&TRUPML,DISP=(OLD,DELETE)
//PAC7GL       DD DSN=&&TRUPGL,DISP=(OLD,DELETE)
//SORTWK01     DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02     DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03     DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//PAC7ET   DD SYSOUT=&OUT
```

3.2.2. TRJC: COMPRESSION OF ARCHIVED JOURNAL
3.2.2.1. TRJC: INTRODUCTION

TRJC: INTRODUCTION

From the VisualAge Pacbase archived Journal, the TRJC procedure produces a compressed Journal containing only effective, or useful, transactions, by eliminating the intermediary transactions which are known to be useless for the transfer.

User input may include an interval of dates and/or session numbers in order to limit transfer processing to the archived Journal's transactions belonging to that interval only.
If there is no optional user input, the compression is carried out on the complete archived Journal.

Also, you have the possibility to erase user codes and/or Change numbers from the archived Journal. As a result, a higher rate of compression is obtained.

In this case, transfer criteria based on user codes and Changes can no longer be used.

NOTES:

For technical reasons, the TRJC procedure should not be used when the archived Journal includes batch update transactions.

As a result, Pac/transfer updates -- performed in batch mode -- should not belong to an archived Journal to be used for another transfer, if the initial target environment becomes the new source environment.

Journal compressing is not required, it depends on the site's requirements (Journal volume, frequency of transfer operations, etc).

EXECUTION CONDITION

None.

RESULT

A smaller archived Journal including effective transactions only.

OUTPUT REPORT

Statistical data on the TRJC execution.

3.2.2.2. TRJC: USER INPUT

TRJC: USER INPUT

. User identification line (required)

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

. Options

Pos.	Len.	Val.	Meaning
1	1		Deletion of user codes:
		'0'	Yes
		'1'	No
2	1		Deletion of Change numbers:
		'0'	Yes
		'1'	No
3	4		Start session number
7	4		End session number
11	8		Start date in the form CCYYMMDD
19	8		End date in the form CCYYMMDD

3.2.2.3. TRJC: DESCRIPTION OF STEPS

TRJC: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

DELETE/DEFINE OF COMPRESSED JOURNAL FILE: IDCAMS

COMPRESSION (FIRST STAGE): PTUG05

```
.Permanent input files:
 -Sequential journal
  PAC7PJ:  DSN=&INDEXQ..&ROOT.&FILE.PJ
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Transaction file:
 -User input
  PAC7MB:  DSN=&&TRJCMB

.Output file:
 -Temporary journal
  PAC7GP:  DSN=&&PAC7GP

.Output reports:
 -Check on input:
  PAC7ET
 -Batch procedure abend report
  PAC7DD

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03
```

COMPRESSION (SECOND STAGE): PTUG06

.Input transaction file:
 -Temporary file
  PAC7GP: DSN=&&PAC7GP

.Output file:
 -Sequential compressed file
  PAC7PK: DSN=&&PAC7PK

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03

CLASSIFICATION OF DELETIONS/CREATIONS: PTUG07

.Input file:
 -Index file
  PAC7AN: DSN=&INDEX..&ROOT.&FILE.AN

.Input transaction files:
 -Temporary journal
  PAC7PK: DSN=&&PAC7PK

.Output file:
 -Compressed sequential file
  PAC7PL: DSN=&INDEXP..&ROOT.&FILE.JT

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03

3.2.2.4. TRJC: EXECUTION JCL

```
//*******************************************************************
//* PAC/TRANSFER    :    COMPRESSION OF THE ARCHIVED JOURNAL          *
//*******************************************************************
//$RADP.TRJC   PROC FILE=$FILE, VA PAC PHYSICAL DATABASE NUMBER
//         ROOT=$ROOT,          RADICAL OF THE VA PAC SYSTEM
//         INDEX='$INDEX',      INDEX VA PAC VSAM FILES
//         INDEXP='$INDEXP',    INDEX OF VA PAC NON-VSAM FILES
//         INDEXQ='$INDEXQ',    INDEX OF VA PAC ARCHIVED JOURNAL
//         UNITS=$UNITO,        COMPRESSED JOURNAL FILE UNIT
//         VOLS='SER=$VOLO',    COMPRESSED JOURNAL FILE UNIT
//*:       VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//         OUT=$OUT,            OUTPUT CLASS
//         STEPLIB='$MODB',     LOAD-MODULE LIBRARY
//         PSBLIB='$PSBLIB',    PSB LIBRARY
//         DBDLIB='$DBDLIB',    DBD LIBRARY
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         UWK=$UWK,            WORK UNIT
//    SPAJT='(TRK,(30,10))',    SPACE OF THE 'FH' FILE
//    SPAGP='(TRK,(100,10),RLSE)', SPACE OF THE 'GP' FILE
//    SORTLIB='$BIBT',          SORT LIBRARY
//    CYL=(10,2),               SIZE OF SORT-WORK FILES
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TRJCMB,DISP=(,PASS),UNIT=&UWK,
//         SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//DELTJ   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DL&ROOT.&FILE.JT),DISP=SHR
//*
//PTUG05  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG05,PTUG05$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF   DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7PJ       DD DSN=&INDEXQ..&ROOT.&FILE.PJ(0),DISP=OLD
//PAC7MB       DD DSN=&&TRJCMB,DISP=(OLD,DELETE)
//PAC7ET    DD SYSOUT=&OUT
//PAC7DD    DD SYSOUT=&OUT
//PAC7GP    DD DSN=&&PAC7GP,DISP=(,PASS),UNIT=&UWK,
```

```
//          DCB=(RECFM=FB,LRECL=223,BLKSIZE=26760),
//          SPACE=&SPAGP
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//*
//PTUG06  EXEC PGM=PTUG06,COND=(0,NE,PTUG05)
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7GP   DD DSN=&&PAC7GP,DISP=(OLD,PASS)
//PAC7PK   DD DSN=&&PAC7PK,DISP=(,PASS),UNIT=&UWK,
//         DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179),
//         SPACE=&SPAJT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PTUG07  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG07,PTUG07$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(0,NE,PTUG05)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AN$SUF   DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7PK       DD DSN=&&PAC7PK,DISP=(OLD,PASS)
//PAC7PL   DD DSN=&INDEXP..&ROOT.&FILE.JT,
//         DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179),
//          SPACE=&SPAJT
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
```

3.2.3. TRPF: TRANSFER-FILE CREATION
3.2.3.1. TRPF: INTRODUCTION

TRPF: INTRODUCTION

From the archived Journal --whether compressed or not, depending on the site's choice and according to the contents of the Parameter file-- the TRPF procedure produces a Transfer file, which has the following characteristics:

1. The only transactions processed are those meeting the source selection parameters (sessions, Libraries, users, Changes),

2. The values of the selected parameters are replaced by those of the target parameters specified in the Parameter file,

3. The selected transactions of the archived journal are duplicated as many times as there are target session numbers and target Library codes.

The file may contain the transactions for one, several or all of the Sets.

EXECUTION CONDITIONS

None.

RESULT

The TRPF procedure produces a Transfer file, which will be used by the TRRP procedure.

3.2.3.2. TRPF: USER INPUT

TRPF: USER INPUT

. User identification line (required)

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

. Transaction Set for processing selection line (required)

Pos.	Len.	Value	Meaning
2	2	'LT'	
4	5	llll	Transaction Set for processing code
		'*****'	Selection of all Sets

NOTE: The selection of all Sets necessarily implies that only one LT-type line be entered (with the value '*****' in Positions 4 to 8).

3.2.3.3. TRPF: DESCRIPTION OF STEPS

TRPF: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

DELETE/DEFINE OF TRANSFER FILE: IDCAMS

CREATION OF TRANSFER FILE: PTUG50

```
.Permanent input files:
 -Index file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -Parameter file
  PAC7UV:  DSN=&INDEX..&ROOT.&FILE.UV
 -Sequential or compressed file
  PAC7JT:  DSN=&INDEXP..&ROOT.&FILE.JT

.Transaction file:
 -User input
  PAC7MB:  DSN=&&TRPFMB

.Output files:
 -Sequential transfer journal
  PAC7TJ:  DSN=&INDEXP..&ROOT.&FILE.TJ

.Sort file(s):
  SORTWK01
  SORTWK02
  SORTWK03

.Output reports:
 -Transfer statistics
  PAC7ET
 -Check on user
  PAC7DD
 -TRPF-transaction list
  PAC7ER
```

3.2.3.4. TRPF: EXECUTION JCL

```
//*******************************************************************
//* PAC/TRANSFER    :     CREATION OF THE TRANSFER FILE
//*******************************************************************
//$RADP.TRPF   PROC FILE=$FILE, VA PAC PHYSICAL-DATABASE NUMBER
//         ROOT=$ROOT,          RADICAL OF THE VA PAC SYSTEM
//         INDEX='$INDEX',      INDEX OF VA PAC VSAM FILES
//         INDEXP='$INDEXP',    INDEX OF VA PAC NON-VSAM FILES
//         UNITS=$UNITO,        NON-VSAM USER FILE UNIT
//         VOLS='SER=$VOLO',    NON-VSAM USER FILE VOLUME
//*:       VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//         OUT=$OUT,            OUTPUT CLASS
//         STEPLIB='$MODB',     LOAD-MODULE LIBRARY
//         PSBLIB='$PSBLIB',    PSB LIBRARY
//         DBDLIB='$DBDLIB',    DBD LIBRARY
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         UWK=$UWK,            WORK UNIT
//    SPATJ='(TRK,(30,10))',    SPACE OF THE 'FH' FILE
//    SORTLIB='$BIBT',          SORT LIBRARY
//    CYL=5,                    SIZE OF SORT-WORK FILES
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TRPFMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,1),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//DELTJ   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DL&ROOT.&FILE.TJ),DISP=SHR
//*
//PTUG50  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG50,PTUG50$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7JT   DD DSN=&INDEXP..&ROOT.&FILE.JT,DISP=SHR
//PAC7MB   DD DSN=&&TRPFMB,DISP=(OLD,DELETE)
//PAC7UV   DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7ET   DD SYSOUT=&OUT
//PAC7ER   DD SYSOUT=&OUT
//PAC7TJ   DD DSN=&INDEXP..&ROOT.&FILE.TJ,
```

```
//         DISP=(,CATLG),UNIT=&UNITS,
//         VOL=&VOLS,
//         DCB=(RECFM=FB,LRECL=172,BLKSIZE=8600),
//         SPACE=&SPATJ
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP  DD SYSOUT=&OUT
```

3.2.4. TRDU: DSMS-ENVIRONMENT PREPARATION
3.2.4.1. TRDU: INTRODUCTION

TRDU: INTRODUCTION

The DSMS-Environment Preparation procedure (TRDU) must be used when the VisualAge Pacbase Database is under DSMS control, and when source criteria include a selected Change number.

NOTE: TRDU can operate for either one or all of the Sets defined in the Parameters file.

The VisualAge Pacbase authorizations notified for the target Change(s) must include the authorizations of the source Change(s). Otherwise, transfers in VA Pac will be rejected.

Compliance to this requirement is ensured by the TRDU procedure which temporarily aligns the target Change(s) with the source Changes regarding their VisualAge Pacbase authorizations.

NOTE: When source criteria do not include a selected Change number, TRDU cannot be applied because of the bulk of Changes involved. In this case, manual checks and alignments will be necessary.

TRDU takes into account the following additional parameters:

. If the Parameters file specifies the transfer of transactions from one source Library to one or more target Libraries, the target Change must authorize the transactions of the target Library(ies).

. If the Parameters file specifies the transfer of transactions from one source user to a target user, the target Change number must authorize the transactions under this target user code.

The TRDU procedure produces two files:

1. A DSMS update-transaction file to allow target Change(s) to accept updates made on the source Change(s).

>>> Also, all VA Pac authorizations attached to source Changes are withdrawn. This means that during the transfer operation, no update made in VA Pac in relation to those Changes will be allowed.

This update must be executed BEFORE the transfer operation.

2. A DSMS update transactions file to set the authorizations of the source and target Changes to their initial state.

   This update must be executed AFTER the transfers are introduced in the VA Pac Database.

EXECUTION CONDITION

None.

RESULT

Two DSMS batch update-transaction files, one of which should be applied before the transfers, the other after all transfers.

3.2.4.2. TRDU: USER INPUT

TRDU: USER INPUT

. User identification line (required)

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

. TRANSACTION SET selection line (required)

Pos.	Len.	Value	Meaning
2	2	'LT'	
4	5	llll	Selected Transaction Set code
		'*****'	Selection of all Sets

One and only one LT-type line is required.

3.2.4.3. TRDU: DESCRIPTION OF STEPS

TRDU: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

SELECTION OF SETS: PTUG42

.Input files:
 -Data file
  PAC7AR: DSN=&INDEX..&ROOT.&FILE.AR
 -Error-messages file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE
 -Parameter file
  PAC7UV: DSN=&INDEX..&ROOT.&FILE.UV
 -User input
  PAC7MB: DSN=&&TRDUMB

.Output file:
 -SETS file
  PAC7BM: DSN=&&TRDUBM

.Output reports:
 -Check on user
  PAC7DD
 -Check on extraction
  PAC7ET

PREPARATION OF DSMS BEFORE TRANSFERS: PTUG44

.Input files:
 -Parameter file
  PAC7UV: DSN=&INDEX..&ROOT.&FILE.UV
 -Error-message file
  PAC7AE: DSN=&INDEX.&ROOT.&ROOT.AE
 -Data file
  PAC7AR: DSN=&INDEX.&ROOT.&FILE.AR
 -VisualAge Pacbase element file
  PACDDC: DSN=INDEXD..&ROOTD.&FILED.DC
  PACDD3: DSN=INDEXD..&ROOTD.&FILED.D3
 -Batch-transaction file
  PAC7MB: DSN=&&TRDUBM

.Output files:
 -Source/target initial-state creation transactions
  PAC7CI: DSN=&&TRDUCI
 -Source/target initial-state deletion transactions
  PAC7SI: DSN=&&TRDUSI
 -Target-change authorizations Preparation file
  PAC7GC: DSN=&&PAC7GC

.Output report:
 -Execution report
  PAC7ET

GENERATION OF TARGET CHANGE TRANSACTIONS: PTUG46

.Input files:
 -Error-message file
  PAC7AE: DSN=&INDEX.&ROOT.&ROOT.AE
 -Data file
  PAC7AR: DSN=&INDEX.&ROOT.&FILE.AR

-Preparation file for target-Change authorizations
PAC7GC: DSN=&&TRDUGC

.Output files:
-Target before-transfer creation transactions
PAC7CC: DSN=&&TRDUCC
-Target after-transfer deletion transactions
PAC7SC: DSN=&&TRDUSC

.Sort file:
SORTWK01
SORTWK02
SORTWK03

.Output report:
-Execution report
PAC7ET

CREATION OF BEFORE-TRANSFER TRANSACTIONS: IEBGENER

.Input files:
-Initial-state deletion file
SYSUT1: DSN=&&TRDUSI
-Before-transfer target creation file
SYSUT1: DSN=&&TRDUCC

.Output file:
-Transactions for DUPT before VisualAge Pacbase update
SYSUT2: DSN=&&TRDUAV

CREATION OF AFTER-TRANSFER TRANSACTION FILE: IEBGENER

.Input files:
-Initial report creation file
SYSUT1: DSN=&&TRDUCI
-After-transfer target deletion file
SYSUT1: DSN=&&TRDUSC

.Output files:
-Transactions for DUPT after VisualAge Pacbase update
SYSUT2: DSN=&&TRDUAP

3.2.4.4. TRDU: EXECUTION JCL

```
//*******************************************************************
//* PAC/TRANSFER     :    PREPARATION OF THE DSMS ENVIRONMENT
//*******************************************************************
//$RADP.TRDU   PROC FILE=$FILE, VA PAC PHYSICAL-DATABASE NUMBER
//         FILED=$FILED,        DSMS PHYSICAL-DATABASE NUMBER
//         ROOT=$ROOT,          ROOT OF THE VA PAC SYSTEM
//         ROOTD=$ROOTD,        ROOT OF THE DSMS SYSTEM
//         INDEX='$INDEX',      INDEX OF VA PAC VSAM FILES
//         INDEXP='$INDEXP',    INDEX OF VA PAC NON-VSAM FILES
//         INDEXD='$INDEXD',    INDEX OF THE 'DC' FILE
//*:       VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//         OUT=$OUT,            OUTPUT CLASS
//         STEPLIB='$MODB',     LOAD-MODULE LIBRARY
//         PSBLIB='$PSBLIB',    PSB LIBRARY
//         DBDLIB='$DBDLIB',    DBD LIBRARY
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         UWK=$UWK,            WORK UNIT
//         SPACI='(TRK,(90,15),RLSE)', SPACE  CI FILE
//         SPACC='(TRK,(90,15),RLSE)', SPACE  CC FILE
//         SPAAV='(TRK,(90,15),RLSE)', SPACE  AV FILE
//         SPAAP='(TRK,(90,15),RLSE)', SPACE  AP FILE
//         SPAGC='(TRK,(30,10))', SPACE OF THE 'FH' FILE
//         SORTLIB='$BIBT',     SORT LIBRARY
//         CYL=2,               SIZE OF SORT-WORK FILES
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TRDUMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//PTUG42  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG42,PTUG42$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV       DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7MB   DD DSN=&&TRDUMB,DISP=(OLD,DELETE)
//PAC7BM   DD DSN=&&TRDUBM,DISP=(,PASS),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE)
//PAC7DD   DD SYSOUT=&OUT
//PAC7ET   DD SYSOUT=&OUT
```

```
//*
//PTUG44  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG44,PTUG44$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7DC$SUFD  DD DSN=&INDEXD..&ROOTD.&FILED.DC,DISP=SHR
//PAC7D3$SUFD  DD DSN=&INDEXD..&ROOTD.&FILED.D3,DISP=SHR
//PAC7UV       DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7MB   DD DSN=&&TRDUBM,DISP=SHR
//PAC7CI   DD DSN=&&TRDUCI,DISP=(,PASS),UNIT=&UWK,
//         DCB=(RECFM=FB,BLKSIZE=27500,LRECL=250),
//         SPACE=&SPACI
//PAC7SI   DD DSN=&&TRDUSI,DISP=(,PASS),UNIT=&UWK,
//         DCB=(RECFM=FB,BLKSIZE=27500,LRECL=250),
//         SPACE=&SPACI
//PAC7GC   DD DSN=&&TRDUGC,DISP=(,PASS),UNIT=&UWK,
//         DCB=(RECFM=FB,BLKSIZE=32000,LRECL=40),
//         SPACE=&SPAGC
//PAC7ET   DD SYSOUT=&OUT
//*
//PTUG46  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG46,PTUG46$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT  DD SYSOUT=&OUT
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//         DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7GC   DD DSN=&&TRDUGC,DISP=SHR
//PAC7CC   DD DSN=&&TRDUCC,DISP=(NEW,PASS),
//         UNIT=&UWK,SPACE=&SPACC,
//         DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
```

```
//PAC7SC   DD DSN=&&TRDUSC,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=&SPACC,
//          DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
//PAC7ET    DD SYSOUT=&OUT
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//*
//CREAV     EXEC PGM=IEBGENER
//SYSOUT    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//SYSUT1    DD DSN=&&TRDUSI,DISP=SHR
//          DD DSN=&&TRDUCC,DISP=SHR
//SYSUT2    DD DSN=&&TRDUAV,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=&SPAAV,
//          DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
//CREAP     EXEC PGM=IEBGENER
//SYSOUT    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//SYSUT1    DD DSN=&&TRDUSC,DISP=SHR
//          DD DSN=&&TRDUCI,DISP=SHR
//SYSUT2    DD DSN=&&TRDUAP,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=&SPAAP,
//          DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
===SEQ FOR S2K
```

3.2.5. UPDATE OF DSMS FUNCTION BEFORE VA PAC UPDATE

UPDATE OF DSMS BEFORE VA PAC UPDATE

This update is performed using, as input of the DUPT procedure, the first file produced by the DSMS authorization update process.

3.2.6. TRRP: GENERATION OF TRANSFER TRANSACTIONS
3.2.6.1. TRRP: INTRODUCTION

TRRP: INTRODUCTION

Once the Transfer file has been built, the TRTP procedure generates transfer transactions. These have the same format as batch update transactions applicable in VA Pac by the UPDT procedure.

The transaction generation may be performed on the whole of the Transfer file or on selected parts, based on the following criteria:

1. Transaction Set (required),
2. Target Session.

Values for both criteria are indicated on the user identification line '*'. Sort options are also available and must be entered in a J-type line.

Each combination of criteria corresponds to a TRRP execution mode.

1. Standard execution mode (by Transaction Set):

. Transaction Set code different from '*****'
. Absence of target session

TRRP considers transactions that belong to the selected Transaction Set only. Since you have not selected a target session, transactions are generated for all target sessions found in the Parameters file regarding this Set.

However, you must run as many TRRP executions as there are target sessions:

A specific attribute -- SESSION PROCESSED -- is automatically positioned in the Parameter file once all transactions have been generated for a given session.

As a result, if this attribute is positioned for a given session (see also the other execution modes, described in Paragraphs 2 and 3), transactions for that session will not be generated and TRRP will automatically proceed with the next target session, as listed in the Parameter file.

This execution mode brings an automatic control over your transfer operations since it avoids duplicating transactions which could otherwise happen when prior TRRP executions have been run.

The TRRP standard execution mode is therefore recommended for sites where Pactransfer operations involve large volumes of transactions.

A Warning message will tell you when all sessions have been dealt with.

Generated transactions must then be used by the VisualAge Pacbase batch update procedure (UPDT).

You may prefer to concatenate all TRRP subsequent outputs and run the UPDT procedure only once.

2. Execution mode by Session:

. Transaction Set code different from '*****'
. Target session: 'nnnnT' or '*****'

TRRP considers transactions that belong to the selected Transaction Set only.

1. If you have selected a target session, transactions are generated for this session only.

2. If you have selected all sessions ('*****'), transactions are systematically generated for all target sessions, all in one TRRP execution.

>>>> A specific attribute -- SESSION PROCESSED -- is automatically positioned in the Parameters file once all transactions have been generated for a given session.

Generated transactions must then be used by the VA Pac batch update procedure (UPDT).

3. Execution mode for all Sets and all target sessions:

. Transaction Set code: '*****'
. Target session number: '****'

Transactions are systematically generated for all Sets and for all their respective target sessions.

>>>> A specific attribute -- SESSION PROCESSED -- is automatically positioned in the Parameters file once all transactions have been generated for a given session.

Generated transactions must then be used by the VA Pac batch update procedure (UPDT).

EXECUTION CONDITIONS

The Transfer file must exist (created by the TRPF procedure). Authorization level 4 is required to run a TRRP execution.

RESULT OBTAINED

Transfer transactions formatted for the VA Pac UPDT batch update procedure.

3.2.6.2. TRRP: USER INPUT

TRRP: USER INPUT

. User identification line (required)

Pos.	Len.	Value	Significance
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	5		Selection of target session(s):
		blank	. All target sessions (default), one session processed per TRRP execution. This value cannot be used when all Transaction sets are selected
		nnnnT	. Target session number (required)
		'*****'	. All target sessions processed in one TRRP execution
40	5		Selection of Transaction Set(s):
		lllll	Transaction Set code
		'*****'	All Transaction Sets

. Sort Options line

Pos.	Len.	Value	Significance
2	1	'J'	Line code
4	1	' '	Chronological list
		'N'	No chronological list
5	1	' '	List by user
		'N'	No list by user
6	1	' '	List by library
		'N'	No list by library

3.2.6.3. TRRP: DESCRIPTION OF STEPS

TRRP: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

PREPARATION OF EXTRACTION: PTUG60

.Permanent input files:
 -Index file
 PAC7AR: DSN=&INDEX..&ROOT.&FILE.AR
 -Error messages
 PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE
 -Parameter-setting file
 PAC7UV: DSN=&INDEX..&ROOT.&FILE.UV
 -Compressed journal file
 PAC7TJ: DSN=&INDEX..&ROOT.&FILE.TJ

.Transaction file:
 -User input
  PAC7MB : DSN=&&TRRPMB

.Output file:
 -Parameter-line file
  PAC7BM : DSN=&&PACXMB
 -Temporary journal file
  PAC7PJ : DSN=&&PACXPJ

.Output reports:
 -Transfer statistics
  PAC7ET
 -User check
  PAC7DD

.Return code(s):
 - 4 if no more session to extract

INITIALIZATION OF KSDS WORK FILE: MAXKEY

EXTRACTION: PACX

This step extracts transactions based on user input.

.Permanent input files:
 -Data file
  PAC7AR: DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN: DSN=&INDEX..&ROOT.&FILE.AN
 -Error-message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE
 -Transactions selected on Journal
  PAC7PJ: DSN=&&PACXPJ

.Input transaction file:
 -User input
  PAC7MB: DSN=&&PACXMB

.Work files
 -User input
  PAC7BM: DSN=&&PACXBM
 -Journal transactions (EXPJ)
  PAC7MJ: DSN=&&PACXMJ
 -Extracted transactions
  PAC7WD: DSN=&&PACXWD

.Output file:

```
 -Transactions extracted for UPDT
  PAC7MV:  DSN=&&MV

.Sort file(s):
  SORTWK01
  SORTWK02
  SORTWK03

.Output reports:
 -General program-stream printout
  PAC7IA
 -List of errors on input transactions
  PAC7DD
 -Extraction list report(s)
  PAC7EE
  PAC7EP
  PAC7EQ
  PAC7EZ



.Return code(s):
  0:  No error
  8:  Serious error (detailed in PAC7DD)
```

POSITIONNING THE 'PROCESSED SESSION' ATTRIBUTE: PTUG61

```
.Permanent input files:
 -Index file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Error-message file
  PAC7AE:  DSN=&INDSV..&ROOT.&ROOT.AE

.Input transaction file
 -User input
  PAC7MB:  DSN=&&PACXMB

.Input/Output file:
 -Parameter-settings
  PAC7UV:  DSN=&INDEX..&ROOT.&FILE.UV

.Output report(s):
 -Transfer statistics
  PAC7ET
```

3.2.6.4. TRRP: EXECUTION JCL

```
//*****************************************************************
//* PAC/TRANSFER  -  EXTRACTION FROM THE COMPRESSED JOURNAL FILE      *
//*****************************************************************
//$RADP.TRRP   PROC FILE=$FILE, VA PAC PHYSICAL-DATABASE NUMBER
//         ROOT=$ROOT,          ROOT OF THE VA PAC SYSTEM
//         INDEX='$INDEX',      INDEX OF VA PAC VSAM FILES
//         INDEXP='$INDEXP',    INDEX OF VA PAC NON-VSAM FILES
//*:       VSAMCAT='$VCAT',     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',     SYSTEM VSAM CATALOG
//         OUT=$OUT,            OUTPUT CLASS
//         STEPLIB='$MODB',     LOAD-MODULE LIBRARY
//         PSBLIB='$PSBLIB',    PSB LIBRARY
//         DBDLIB='$DBDLIB',    DBD LIBRARY
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         UWK=$UWK,            WORK UNIT
//         SPAMV='(TRK,(30,10))', SPACE OF THE 'MV' FILE
//         SPAPJ='(TRK,(30,10))', SPACE OF THE SELECTED 'PJ' FILE
//         SPAMB='(TRK,(5,1),RLSE)', SPACE OF THE COMMAND FILE
//         SPATD='(TRK,(50,10),RLSE)', SPACE OF THE CPSN FILE
//         SORTLIB='$BIBT',       SORT LIBRARY
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&TRRPMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,1),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//VERIFY EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//DDAR     DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//PTUG60  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG60,PTUG60$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
```

```
//PAC7UV       DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7JT       DD DSN=&INDEXP..&ROOT.&FILE.TJ,DISP=SHR
//PAC7MB   DD DSN=&&TRRPMB,DISP=(OLD,DELETE)
//PAC7BM   DD DSN=&&PACXMB,DISP=(,PASS),UNIT=SYSDA,
//            SPACE=(TRK,(1,1)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
//PAC7PJ   DD DSN=&&PACXPJ,DISP=(,PASS),UNIT=SYSDA,
//            SPACE=&SPAPJ,
//            DCB=(RECFM=FB,LRECL=167,BLKSIZE=16700)
//PAC7ET   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
//*
//MAXKEY  EXEC PGM=IDCAMS
//**********************
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&&SYSEXT,DISP=(NEW,KEEP),
//         SPACE=(CYL,(3,3)),
//         LRECL=112,RECORG=KS,KEYOFF=0,KEYLEN=12
//MAXKEY   DD DSN=&INDEXP..&ROOT.&ROOT.SY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(REPRO999),DISP=SHR
//*
//PACX    EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACX,PACX$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//******** FICHIERS BASE
//PAC7AE$SUF  DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF  DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF  DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7PJ   DD DSN=&&PACXPJ,DISP=(OLD,DELETE)
//******** ETATS
//PAC7IA   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
//PAC7EE   DD SYSOUT=&OUT
//PAC7EP   DD SYSOUT=&OUT
//PAC7EQ   DD SYSOUT=&OUT
//PAC7EZ   DD SYSOUT=&OUT
//******** FICHIERS DE TRAVAIL
//SYSEXT   DD DSN=&&SYSEXT,DISP=(OLD,DELETE,DELETE)
//PAC7MB   DD DSN=&&PACXMB,DISP=(OLD,PASS)
//PAC7BM   DD DSN=&&PACXBM,DISP=(,PASS),UNIT=&UWK,
//            DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7MM   DD DUMMY
//PAC7MJ   DD DSN=&&PAC7MJ,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMV,
//            DCB=(RECFM=FB,LRECL=152,BLKSIZE=6080)
//PAC7TE   DD DUMMY
//PAC7RE   DD DUMMY
//PAC7RM   DD DUMMY
//PAC7WD   DD DSN=&&PACXWD,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPATD,
//            DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
//******** FICHIERS DE TRI
```

```
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//******** FICHIERS EN SORTIE
//PAC7MV   DD DSN=&&MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMV,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MR   DD DUMMY
//PAC7TD   DD DUMMY
//PAC7GY   DD DUMMY
//PAC7UE   DD DUMMY
//*
//PTUG61  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUG61,PTUG61$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=((0,NE,PACX),(0,NE,PTUG60))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV       DD DSN=&INDEX..&ROOT.&FILE.UV,DISP=SHR
//PAC7MB       DD DSN=&&PACXMB,DISP=(OLD,DELETE)
//PAC7ET       DD SYSOUT=&OUT
//
```

3.2.7. UPDATE OF THE VISUALAGE PACBASE DATABASE

UPDATE OF THE VISUALAGE PACBASE DATABASE

The VisualAge Pacbase Database is updated via the UPDT procedure, taking the Transfer file -- created by the TRRP procedure -- as input.

In the case of a 'standard processing' of the generation of transfer transactions (see previous subchapter), the following procedures may be executed several times:

. TRRP (Generation of transfer transactions),

. UPDT (Update of the VA Pac Database).

3.2.8. REINITIALIZATION OF THE DSMS ENVIRONMENT

REINITIALIZATION OF THE DSMS ENVIRONMENT

This procedure resets update authorizations on the selected source and target Changes as they were before the transfer operation.

This initial state is obtained by running the DSMS update procedure (DUPT), using as input transactions the contents of the file resulting from the DSMS Environment Preparation procedure (TRDU).

4. MANAGER'S UTILITIES

4.1. STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES

4.1.1. STOP: INTRODUCTION

STOP: INTRODUCTION

The purpose of the STOP procedure is to enhance system efficiency by promoting optimal storage of data when data from large volume databases is distributed among several volumes.

The standard organization of VA Pac data consists of storing more than 80 percent of the most widely used data at the top of a file (in the case of normal operations carried out by a development team). Distribution of data on several volumes therefore has a limited impact on system performance.

Through the STOP procedure, the physical allocation of data on several volumes is optimized -- the most widely used data is distributed on several volumes -- in order to avoid disk contention problems. Thus, performance is also improved.

EXECUTION CONDITION

None, since the database is not updated directly.

4.1.2. STOP: USER INPUT

STOP: USER INPUT

.One command line:

POS.	LEN.	VALUE	MEANING
2	1	' '	Blank line code
3	8	nnnnnnnn	Number of data (including gaps) See back-up statistics on input to the procedure
11	8	ggggggggg	Number of disks
19	8	pppppppp	Number of records per data block

CALCULATION OF THE NUMBER OF RECORDS PER DATA BLOCK

Let N be the number of records per CI.
Use a multiple of N.

Example: CI = 4096K
CI = 4096K (25 AR records per CI)
Let's use a data block of 100 CI.
Therefore the number of records per block is 2500.

With nnnnnnn given to be divided among 3 volumes, the command line looks like the following:
__nnnnnnn_____3____2500

PRINTED OUTPUT

The STOP procedure prints out a report of the resulting storage.

RESULT

The result of the STOP procedure is a standard back-up (PC) of the database in which data storage has been carried out according to the user input command. This data storage must be retrieved as input to the standard Restoration procedure (REST), which provides for the multi-volume allocation of the Data file.

4.1.3. STOP: DESCRIPTION OF STEPS

STOP: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

STORAGE OPTIMIZATION OF DATA: PTUR00

```
.Permanent input files:
 -Sequential image of the database
  PAC7PC:  DSN=&INDEXQ..&ROOT.&FILE.PC(0)
If backup Dispatch option :
 -Sequential image #2 of the database
  PAC7PD:  DSN=&INDEXQ..&ROOT.&FILE.PD(0)

.Input transaction file:
 -User transaction
  PAC7MB:  DSN=&&STOPMB

.Output file:
 -Sequential image of the database
  PAC7CP:  DSN=&INDEXQ..&ROOT.&FILE.PC(+1)
If Dispatch backup option :
 -Sequential image #2 of the database
  PAC7DP:  DSN=&INDEXQ..&ROOT.&FILE.PD(+1)

.Sort files:
  SORTWK01
  SORTWK02
  SORTWK03

.Output report:
 -Execution report
  PAC7EU
```

4.1.4. STOP: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : STORAGE OPTIMIZATION OF VA Pac DATA               *
//*                 (FOR MULTI VOLUMES 'AR' DATABASE)                *
//*********************************************************************
//$RADP.STOP PROC FILE=$FILE,     NUMBER OF PHYSICAL DATABASE
//         ROOT=$ROOT             ROOT OF THE VA PAC SYSTEM
//         INDEXQ='$INDEXQ',      INDEX OF DATA GROUP FILES
//         OUT='$OUT',            OUTPUT CLASS
//         CYL=10,                SORT WORKS SIZE
//         SPAPC='(TRK,(100,10),RLSE)',  BACKUP SPACE
//         SPAPD='(TRK,(100,10),RLSE)',  BACKUP SPACE 2
//         STEPLIB='$MODB',       LIBRARY OF BATCH LOAD-MODULES
//         SORTLIB='$BIBT',       SORT LIBRARY
//         UWK=$UWK,              WORK FILE UNIT
//         VOLS='SER=$VOLO',      BACKUP FILE VOLUME
//         UNITS=$UNITO           BACKUP FILE UNIT
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&STOPMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//PTUR00  EXEC PGM=PTUR00
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7MB   DD DSN=&&STOPMB,DISP=(OLD,DELETE)
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(0),DISP=SHR
//PAC7PD   DD DSN=&INDEXQ..&ROOT.&FILE.PD(0),DISP=SHR
//PAC7CP   DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7DP   DD DSN=&INDEXQ..&ROOT.&FILE.PD(+1),DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPD,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7EU   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
```

4.2. SESSION MANAGEMENT

4.2.1. ESES - CSES: INTRODUCTION

ESES - CSES: INTRODUCTION

The VA Pac session number cannot be greater than 9999.

When the session number is close to 9999, the utility program re-assigns all the session numbers, by incrementing the numbers of frozen sessions by 1 (starting from session 0001 or from a session chosen by the Administrator).

NOTE: The freeze is performed by the UPDT procedure. It increments the current session number.

This reassignment is carried out on sequential images of the files that include the session number, i.e. the backup files of the Database (PC), of the Journal (PJ), of the Print-Generation requests (PG), of the Production Environment (PP), of the DSMS Journal (BJ), of the DSMS Database (BB), and of the Pactable Database (TC).

This utility includes two procedures: ESES and CSES.

4.2.2. ESES: EXTRACTION OF SESSION NUMBERS

ESES: INTRODUCTION

The Extraction of Session Numbers procedure (ESES) creates a correspondence-table file linking older frozen sessions and new frozen sessions.

PRELIMINARY OPERATIONS

Backup of the VA Pac files:

```
.Archival of the Journal (ARCH)
.Backup of the VA Pac Database (SAVE)
.Backup of the Generation-Print requests file (SVAG)


If PEI is installed:.PEI backup (SVPE)


If Pactables is installed:.Table backup (SVTA)
```

If DSMS is installed, perform a backup of the DSMS environment, by:

```
.Archiving the DSMS Journal (DARC)
.Backing up the DSMS Database (DSAV)
```

EXECUTION CONDITIONS

None.

Batch procedure access authorization option: level 4 required.

USER INPUT

Batch procedure access authorization option: a '*' line with User code and Password is required.

One line per session number to force :

Pos.	Lon.	Valeur	Meaning
2	1	'S'	Line Code
3	4	nnnn	Original session number
7	4	nnnn	New session number

4.2.3. ESES: DESCRIPTION OF STEPS

ESES: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

CREATION OF THE SESSION-NUMBER CORRESPONDENCE FILE: PTUESS

```
.Permanent input file:
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN

.Input file:
 -Input transactions
  PAC7MB:  DSN=&&ESESMB

.Output file:
 -Session-number correspondence table
  PAC7MV:  DSN=&ESESFILE

.Output reports:
 -Extraction report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

.Return code:
 -8:  No authorization on this batch procedure.
```

4.2.4. ESES: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : SESSION-NUMBER CORRESPONDENCE TABLE                 *
//*********************************************************************
//$RADP.ESES PROC FILE=$FILE,   NUMBER OF THE VA PAC DATABASE
// ROOT=$ROOT,                  ROOT OF THE VA PAC SYSTEM
// INDEX='$INDEX',              INDEX OF VA PAC BACKUP FILES
// ESESFILE=,                   SESSION-NUMBER FILE DSNAME
// VOLS='SER=XXXXXX',           SESSION-NUMBER FILE VOLUME
// UNITS=SYSDA,                 SESSION-NUMBER FILE UNIT
// STEPLIB='$MODB',             LIBRARY OF LOAD-MODULES
// OUT=$OUT,                    OUTPUT CLASS
// INDEXP='$INDEXP',            INDEX OF VA PAC NON VSAM FILES
// PSBLIB='$PSBLIB',            PSB LIBRARY
// DBDLIB='$DBDLIB',            DBD LIBRARY
// RESLIB='$RESLIB',            IMS RESLIB
// PROCLIB='$PRCLIB',           IMS PROCLIB
// UWK=$UWK,                    WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$,IRLM=$IRLM
//*********************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&ESESMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//PTUESS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUESS,PTUESS$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUG DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7MB   DD DSN=&&ESESMB,DISP=(OLD,PASS)
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//PAC7MV   DD DSN=&ESESFILE,DISP=(,CATLG),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=(TRK,(30,5),RLSE),
// DCB=(RECFM=FB,BLKSIZE=5000,LRECL=5)
//SYSUDUMP DD SYSOUT=&OUT
```

4.2.5. CSES: COMPRESSION OF SESSION NUMBERS

CSES: INTRODUCTION

The Compression of Session Numbers procedure (CSES) compresses the session numbers of the VisualAge Pacbase Database logical backups, the Pactables Database if this module is installed on the site, and the DSMS Database if this module is installed on the site. It uses the correspondence table created by the ESES procedure. The resulting files must be restored.

EXECUTION CONDITIONS

None.

However, all the backups to be processed must be valid.

4.2.6. CSES: USER INPUT

CSES: USER INPUT

Batch procedure access authorization: A * line with User Code and Password.

The user input is used to indicate the list of files to be retrieved (PC, PJ, PG, PP, BB, BJ, and TC), in order to execute the retrieval after one or several runs.

The line is built as follows:

Col.	Len.	Value	Meaning
2	1	'S'	Line code
3	21		Code of the files to retrieve (PC PJ PG PP BB BJ TC) separated with a blank
33	4		If the DSMS database has to be retrieved: VA Pac database logical code

4.2.7. CSES: DESCRIPTION OF STEPS

CSES: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

'COMPRESSION' OF SESSION NUMBERS: PTUCSS

```
.Permanent input files:
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input file (from ESES procedure):
 -Session-number correspondence table
  PAC7MV:  DSN=&ESESFILE

.Transaction file:
 -User input
  PAC7MB:  DSN=&&CSESMB

.Retrieval of the VisualAge Pacbase database backup
 -Input
  PAC7PC:  DSN=&INDEXQ..&ROOT.&FILE.PC(0)
  If Dispatch option of the backup:
  PAC7PD:  DSN=&INDEXQ..&ROOT.&FILE.PD(0)
 -Output
  PAC7CP:  DSN=&INDEXQ..&ROOT.&FILE.PC(+1)
  If Dispatch option of the backup:
  PAC7DP:  DSN=&INDEXQ..&ROOT.&FILE.PD(+1)

.Retrieval of the VisualAge Pacbase archived journal:
 -Input
  PAC7PJ:  DSN=&INDEXQ..&ROOT.&FILE.PJ(0)
 -Output
  PAC7JP:  DSN=&INDEXQ..&ROOT.&FILE.PJ(+1)

.Retrieval of the VA Pac generation-print request backup:
 -Input
  PAC7PG:  DSN=&INDEXQ..&ROOT.&FILE.PG(0)
 -Output
  PAC7GP:  DSN=&INDEXQ..&ROOT.&FILE.PG(+1)
```

```
.Retrieval of the PEI backup:
 -Input
  PAC7PP:  DSN=&INDEXQ..&ROOT.&FILE.PP(0)


 -Output
  PAC7EP:  DSN=&INDEXQ..&ROOT.&FILE.PP(+1)
```

If DSMS is installed:

```
.Retrieval of the DSMS database backup:
 -Input
  PACDBB:  DSN=&DSMS..&ROOTD.&FILED.BB(0)
 -Output
  PACDJB:  DSN=&DSMS..&ROOTD.&FILED.BB(+1)

.Retrieval of the DSMS archived journal:
 -Input
  PACDDJ:  DSN=&DSMS..&ROOTD.&FILED.BJ(0)
 -Output
  PAC7JD:  DSN=&DSMS..&ROOTD.&FILED.BJ(+1)
```

If Pactables is installed:

```
.Retrieval of the Pactables database backup:
 -Input
  PACDTC:  DSN=&DSNTC0
 -Output
  PACDCT:  DSN=&DSNTC1

.Output reports:
 -Execution report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD
```

4.2.8. CSES: EXECUTION JCL

```
//************************************************************
//*  VA Pac       : COMPRESSION OF SESSION NUMBERS
//************************************************************
//$RADP.CSES PROC FILE=$FILE,   NUMBER OF THE VA PAC DATABASE
// ROOT=$ROOT,                  ROOT OF THE VA PAC SYSTEM
// INDEX='$INDEX',              INDEX OF VA PAC VSAM FILES
// INDEXQ='$INDEXQ',            INDEX OF VA PAC BACKUP FILES
// ROOTD='$ROOTD',              ROOT OF THE DSMS SYSTEM
// FILED='$FILED',              NUMBER OF THE DSMS SYSTEM
// DSMS='$INDEXQD',             INDEX OF DSMS BACKUP FILES
// DSNTC0=,                     TABLE SAVE (0)
// DSNTC1=,                     TABLE SAVE (+1)
// ESESFILE=,       SESSION-NUMBER FILE'S DSNAME (FROM ESES PROC)
// STEPLIB='$MODB',             LIBRARY OF LOAD-MODULES
// OUT=$OUT,                    OUTPUT CLASS
// VOLS='SER=$VOLO',            VA PAC BACKUP VOLUME
// SPAPC='(CYL,(150,15),RLSE)', VA PAC BACKUP SPACE
// SPAPJ='(TRK,(150,15),RLSE)', VA PAC JOURNAL BACKUP SPACE
// SPAPG='(TRK,(15,5),RLSE)',   VA PAC 'PG' FILE SPACE
// SPAPP='(TRK,(15,5),RLSE)',   VA PAC PEI BACKUP SPACE
// SPABB='(CYL,(150,15),RLSE)', DSMS DATABASE BACKUP SPACE
// SPABJ='(TRK,(150,15),RLSE)', DSMS JOURNAL BACKUP SPACE
// SPATC='(TRK,(150,15),RLSE)', TABLE SPACE
// INDEXP='$INDEXP',            INDEX OF VA PAC NON VSAM FILES
// PSBLIB='$PSBLIB',            PSBs LIBRARY
// DBDLIB='$DBDLIB',            DBDs LIBRARY
// RESLIB='$RESLIB',            IMS RESLIB
// PROCLIB='$PRCLIB',           IMS PROCLIB
// UWK=$UWK,                    WORK UNIT
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$,IRLM=$IRLM
//*****************************************************************
//INPUT  EXEC PGM=PTU001
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&CSESMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PTUCSS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTUCSS,PTUCSS$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUG DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&CSESMB,DISP=(OLD,PASS)
//PAC7MV   DD DSN=&ESESFILE,DISP=SHR
```

```
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(0),DISP=SHR
//PAC7PD   DD DSN=&INDEXQ..&ROOT.&FILE.PD(0),DISP=SHR
//PAC7CP   DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),DISP=(,CATLG),
//    UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPC,
//    DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7DP   DD DSN=&INDEXQ..&ROOT.&FILE.PD(+1),DISP=(,CATLG),
//    UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPC,
//    DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7PJ   DD DSN=&INDEXQ..&ROOT.&FILE.PJ(0),DISP=SHR
//PAC7JP   DD DSN=&INDEXQ..&ROOT.&FILE.PJ(+1),DISP=(,CATLG),
//     UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPJ,
//     DCB=&INDEXQ..DSCB.&ROOT.&FILE.PJ
//PAC7PG   DD DSN=&INDEXQ..&ROOT.&FILE.PG(0),DISP=SHR
//PAC7GP   DD DSN=&INDEXQ..&ROOT.&FILE.PG(+1),DISP=(,CATLG),
//      UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPG,
//      DCB=&INDEXQ..DSCB.&ROOT.&FILE.PG
//PAC7PP   DD DSN=&INDEXQ..&ROOT.&FILE.PP(0),DISP=SHR
//PAC7EP   DD DSN=&INDEXQ..&ROOT.&FILE.PP(+1),DISP=(,CATLG),
//      UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPP,
//      DCB=&INDEXQ..DSCB.&ROOT.&FILE.PP
//PACDBB   DD DSN=&DSMS..&ROOTD.&FILED.BB(0),DISP=SHR
//PACDJB   DD DSN=&DSMS..&ROOTD.&FILED.BB(+1),DISP=(,CATLG),
//      UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABB,
//      DCB=&DSMS..DSCB.&ROOTD.&FILED.BB
//PACDDJ   DD DSN=&DSMS..&ROOTD.&FILED.BJ(0),DISP=SHR
//PAC7JD   DD DSN=&DSMS..&ROOTD.&FILED.BJ(+1),DISP=(,CATLG),
//      UNIT=&UNITS,VOL=&VOLS,SPACE=&SPABJ,
//      DCB=&DSMS..DSCB.&ROOTD.&FILED.BJ
//SYSUDUMP DD SYSOUT=&OUT
//PAC7TC   DD DSN=&DSNTC0,DISP=SHR
//PAC7CT   DD DSN=&DSNTC1,
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPATC,
//         DCB=(RECFM=FB,BLKSIZE=10614,LRECL=1061)
//SYSUDUMP DD SYSOUT=&OUT
```

4.3. GBIR: PARTITIONED DATABASE MANAGER

4.3.1. GBIR: INTRODUCTION

GBIR: INTRODUCTION

The PARTITIONED DATABASE MANAGER (LCU-) is a utility option of the Dictionary function, and its use depends on the corresponding purchase agreement.

Users likely to use this utility are those who work with databases shared by one or more sites, and who might therefore be working on several versions of the same sub-network.

With this utility, you can align all versions of a particular sub-network, taking into account the update transactions performed on any one of these versions.

In more general terms, through the Sub-Network Comparison Utility, any two versions of a sub-network may be aligned. For example, this utility can be used when the current version of a sub-network has to take into account update transactions performed on a frozen session of this sub-network.

For additional information, refer to the OPTIONAL UTILITIES Reference Manual.

PRINCIPLES

Two methods may be used to align a 'slave' sub-network with a 'master' sub-network:

The standard method generates batch transactions which are used to update the 'slave' sub-network. The standard validations performed by the update ensure the consistency of updated data in the 'slave' sub-network.

The second method involves merging the 'master' sub-network with the network containing the 'slave' sub-network: the 'master' sub-network replaces the 'slave' sub-network. The results of the merge must be reorganized via the REOR procedure to obtain a back-up of the new network, which can be used as input to the REST procedure.

No validation is performed on data consistency. Thus, this method must only be used when standard network management ensures data consistency between the networks.

For more details, refer to the OPTIONAL UTILITIES Reference Manual.

1. ALIGNMENT THROUGH THE BATCH UPDATE PROCEDURE

The Sub-Network Comparison Utility generates an update transaction flow making a 'slave' sub-network identical to a 'master' sub-network.

This is done in two steps:

- The extraction, in sequential form, of the sub-network image, which must be aligned via the PACX procedure (EXLI extractor, formatting for CPSN). (For further details, see Chapter STANDARD PROCEDURES, Subchapter 'PACX: Extraction from the VA Pac Database', in the 'Batch Procedures, User's Guide'.)

- The comparison of images, two-by-two, in order to produce an update transaction flow (CPSN procedure).

These two operations may be executed at different sites.

NOTES ON THE GENERATED UPDATE TRANSACTION FLOW

It is logically impossible to align P.I.A.'s: for the modification of a P.I.A. in a 'master' sub-network, the generated update transactions will not be accepted if the P.I.A. is already called in a library of the 'slave' sub-network.

In the update report of the 'slave' sub-network (UPDT procedure), some '0' or 'H' lines may be rejected with the following error message:

"INVALID ABSENCE FOR THE FIELD PROGRAM NAME"

This message can be ignored; the update is executed correctly.

2. ALIGNMENT THROUGH THE SUB-NETWORK MERGE

The Sub-Network Merge Utility generates a sequential file which is the result of the merge of a 'master' sub-network into a target network. This 'master' sub-network completely replaces the 'slave' sub-network.

The replacement of the 'slave' sub-network is done on a library-to-library basis. If the library hierarchy of the 'master' sub-network is different from that of the 'slave' (new, deleted or modified libraries), the modifications must be applied to the target network via the MLIB procedure before the merge procedure.

The library codes may be different in the 'slave' and 'master' sub-networks.

The sub-network merge is executed in three steps:

. Extraction of the 'master' sub-network, whose output is a sequential file (EMSN procedure),

. Merge of the extracted sub-network with the target network (MESN procedure), yielding a merged file to be used as input to the REOR procedure,

. Reorganization of the merge result (REOR procedure), yielding a new network back-up.

These three operations may be executed at different sites.

IMPORTANT NOTE

NO consistency check on the data in the network hierarchy is performed (see paragraph "PRINCIPLES" above).

4.3.2. CPSN: SUB-NETWORK COMPARISON
4.3.2.1. CPSN: INTRODUCTION

CPSN: INTRODUCTION

The Sub-Network Comparison procedure (CPSN) compares the images of two sub-networks extracted by the PACX procedure (EXLI extractor, formatting for CPSN), which may or may not belong to the same network, in order to obtain the batch update transactions which will align the 'slave' sub-network with the 'master' sub-network.

The 'master' sub-network is used as the reference when updating the 'slave' sub-network.

EXECUTION CONDITION

Batch procedure access authorization option: Level 3 is required.

ABENDS

If an abend occurs, the procedure can be restarted as it is once the problem has been solved.

4.3.2.2. CPSN: NOTES ON THE RESULTS

USER INPUT

Batch procedure access authorization option:
One '*'line :

COL.	LEN.	VALUE	MEANING
2	1	*	LINE CODE
3	8	uuuuuuuu	USER CODE
11	8	pppppppp	USER PASSWORD
40	3	ppp	DSMS Product Code
43	6	nnnnnn	DSMS Change number
			(DSMS module only)
49	1		Lock management
		' '	Extract. of locks without user code
		'1'	No extraction of locks
		'2'	Extract. of locks with user code
50	1	' '	No transfer of the password on the * line at the top of generated trans.
		'1'	Transfer of the password on the * line at the top of generated trans.

NOTES ON THE RESULTS

The two sub-networks to be compared must have been extracted via the PACX procedure (EXLI extractor, formatting for CPSN).

They must contain the same number of libraries (checked by the system) and have the same structure.

The comparison is made between libraries located in the same place in the two sub-networks, but it is not necessary for the two corresponding libraries to have the same code.

If the 'master' sub-network contains libraries that do not exist in the 'slave' sub-network, you have to initialize these libraries in the 'slave' sub-network before doing the extraction. To do this, use the MLIB procedure followed by the REST procedure.

4.3.2.3. CPSN: DESCRIPTION OF STEPS

CPSN: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

COMPARISON OF SUB-NETWORKS: PTU850

This step compares two sub-networks with the
same hierarchical structure, one being considered as the
'master', the other as the 'slave'.

.Permanent input file:
 -Error message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

.Transaction file:
 -User input
  PAC7MB: DSN=&&CPSNMB

.Input files from PACX:
 -Master sub-network
  PAC7MA: DSN=&INDEXP..&ROOT.&FILE.&MA
 -Slave sub-network
  PAC7ES: DSN=&INDEXP..&ROOT.&FILE.&SL

.Output file:
 -Update transactions and sort criterion
  PAC7MK

.Output reports:
 -Report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

.Return codes:
 - 0: OK.
 - 8: Error, or no authorization on batch procedure.

SORT OF COMPARISON TRANSACTIONS: SORT

.Input file:
 -Transactions with sort criteria
  SORTIN: DSN=&&PAC7MK

.Output file:
 -Sorted transactions
  SORTOUT: DSN=&&PAC7KM

.Sort criteria: SRTCPSN member of the SY PDS.

.Sort files:
 SORTWK01
 SORTWK02
 SORTWK03

FORMATTING GENERATED TRANSACTIONS: PTU855

This step formats the generated and sorted transactions
and prints them. It is executed when no error is found.

.Permanent input file:
 -Error message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

.Input work file:
 -Sorted generated transactions
  PAC7MK: DSN=&&PAC7KM

.Output file:
 -Transactions generated for update
  PAC7MB: DSN=&&MB

.Output report:
 -Generated transactions
  PAC7EU

4.3.2.4. CPSN: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : PARTITIONED DATABASE MANAGER                      *
//*                  -- SUB-NETWORK COMPARISON --
//*********************************************************************
//$RADP.CPSN PROC FILE=$FILE,    NUMBER OF THE PHYSICAL DATABASE
//         ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//         INDEX='$INDEX',       VSAM INDEX
//         INDEXP='$INDEXP',     INDEX OF SYSTEM NON-VSAM FILES
//         MA=BBM,               SUFFIX OF 'MASTER' SUB NETWORK DSN
//         SL=BBE,               SUFFIX OF 'SLAVE'  SUB NETWORK DSN
//*:       SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//         STEPLIB='$MODB',      LIBRARY OF LOAD-MODULES
//         SORTLIB='$BIBT',      SORT LIBRARY
//         OUT='$OUT',           OUTPUT CLASS
//         UWK=$UWK,             WORK UNIT
//         CYL=10,               SIZE OF SORTWORK
//         SPAMB='(TRK,(10,2),RLSE)',   SPACE OF EXTRACTED TRANSACTIONS
//         PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//         RESLIB='$RESLIB',     IMS RESLIB
//         PROCLIB='$PRCLIB',    IMS PROCLIB
//        BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//        CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&CPSNMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//PTU850  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU850,PTU850$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7MB   DD DSN=&&CPSNMB,DISP=(OLD,PASS)
//PAC7ES   DD DSN=&INDEXP..&ROOT.&FILE.&SL,DISP=OLD
//PAC7MA   DD DSN=&INDEXP..&ROOT.&FILE.&MA,DISP=OLD
//PAC7MK   DD DSN=&&PAC7MK,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300),
//            SPACE=&SPAMB
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT,DCB=BLKSIZE=1330
```

```
//*
//SORTMK  EXEC PGM=SORT,COND=(00,NE,PTU850)
//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)
//SORTIN   DD DSN=&&PAC7MK,DISP=(OLD,DELETE)
//SORTOUT  DD DSN=&&PAC7KM,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300),
//            SPACE=&SPAMB
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(SRTCPSN),DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PTU855  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU855,PTU855$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTU850)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7EU   DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB   DD DSN=&&MB,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6240),
//            SPACE=&SPAMB
//PAC7MK   DD DSN=&&PAC7KM,DISP=(OLD,PASS)
//*
```

4.3.3. SASN: SUB-NETWORK BACKUP
4.3.3.1. SASN: INTRODUCTION

SASN: INTRODUCTION

The Sub-Network Backup procedure (SASN) extracts one or several sub-networks from a database. The result is a consistent set of libraries which will make up a new database (formatted as a backup file to be used as input to the Restoration procedure).

Each extracted sub-network is identified by its lowest-level library; the utility automatically extracts all higher-level libraries pertaining to the sub-network.

The SASN procedure may be equated with the MLIB procedure, the only difference is that the SASN procedure deletes gaps.

EXECUTION CONDITION

The database must be closed to on-line use.

Batch procedure access authorization option: Level 4 is required.

ABNORMAL EXECUTION

If an abend occurs, the procedure may be restarted as it is once the problem has been solved.

4.3.3.2. SASN: USER INPUT

SASN: USER INPUT

Batch procedure access authorization option:
One '*' line with user code and password.

POS.	LEN.	VALUE	MEANING
1	2	' '	Not used
3	3	bbb	Code of lowest-level library of the sub-network to be extracted. (All the upper-libraries of 'bbb' will be automatically extracted.)

The user must code one line per library to be extracted.

4.3.3.3. SASN: DESCRIPTION OF STEPS

SASN: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE VALIDATION: PTU130

This program is always executed.

```
.Permanent input files:
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN

.Transaction input file:
 -Database-selection transactions
  PAC7MB:  DSN=&&SASNMB

.Output files:
 -Sequential data image:
  PAC7RP:  DSN=&&RP  Length=149
  (Must be able to contain all data)
 -Sequential index image
  PAC7NA:  DSN=&&NA  Length=55
  (Must be able to contain all indexes)
 -Sequential frozen data image
  PAC7RA:  DSN=&&RA  Length=149

.Sort file(s):

.Output reports:
 -Execution report
  PAC7DS
 -Batch-procedure authorization option
  PAC7DD
```

.Return codes:

- 0: OK
- 5: At least one of the selected libraries does not exist
- 6: More than 99 libraries are selected
- 8: No authorization on batch procedures.

FORMATTING OF SEQUENTIAL IMAGE: PTU140

This program is executed when no error is found in the input transactions.

.Permanent input files:
-Error message file
PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

.Input work files:
-Data sequential image
PAC7RP: DSN=&&RP
-Index sequential image
PAC7NA: DSN=&&NA
-Frozen data sequential image
PAC7RA: DSN=&&RA

.Output file:
-Database sequential image
PAC7SR: DSN=&INDEXP..&ROOT.&FILE.SN
If Dispatch option:
-Database sequential image ¾2
PAC7PD: DSN=&INDEXP..&ROOT.&FILE.SN2

.Sort file(s):
SORTWK01
SORTWK02
SORTWK03

.Output report:
-Execution report
PAC7DS

4.3.3.4. SASN: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : PARTITIONED DATABASE MANAGER                      *
//*                    -- SUB-NETWORK BACKUP --
//*********************************************************************
//$RADP.SASN PROC FILE=$FILE,    NUMBER OF THE PHYSICAL DATABASE
//         ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//         INDEX='$INDEX',       INDEX OF USER VSAM FILES
//         INDEXP='$INDEXP',     INDEX OF SYSTEM NON-VSAM FILES
//         INDEXQ='$INDEXQ',     DATA GROUP FILE INDEX
//*:       SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:       VSAMCAT='$CATU',      VSAM USER CATALOG
//         STEPLIB='$MODB',      LIBRARY OF LOAD-MODULES
//         SORTLIB='$BIBT',      SORT LIBRARY
//         OUT='$OUT',           OUTPUT CLASS
//         UWK=$UWK,             WORK UNIT
//         CYL='(10,1)',         SIZE OF SORTWORK
//         UNITS='$UNITO',       UNIT OF OUTPUT FILES
//         VOLS='SER=$VOLO',     EXTRACTED SUB-NETWORK VOLUME
//         SPASN='(TRK,(80,10),RLSE)',  SUB-NETWORK BACKUP SPACE
//         PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//         RESLIB='$RESLIB',     IMS RESLIB
//         PROCLIB='$PRCLIB',    IMS PROCLIB
//        BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//        CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//* INPUT :                                                            *
//*                                                                    *
//* COL 01 TO 02 : '  ' (REQUEST)                                      *
//* COL 03 TO 05 : VA PAC LIBRARY CODE TO SELECT                       *
//*                                                                    *
//* RETURN CODES OF THE PROGRAM 'PTU130'                               *
//*                                                                    *
//* - 0000 : OK                                                        *
//* - 0005 : AT LEAST ONE LIBRARY TO BE SELECTED DOES NOT EXIST        *
//* - 0006 : MORE THAN 99 INPUT TRANSACTIONS (LIBRARIES)               *
//*                                                                    *
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC  PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&SASNMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//VERIFY  EXEC  PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PTU130  EXEC  PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU130,PTU130$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
```

```
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7MB   DD DSN=&&SASNMB,DISP=(OLD,DELETE)
//PAC7NA   DD DSN=&&NA,DISP=(,PASS,DELETE),
//            DCB=(RECFM=FB,LRECL=55,BLKSIZE=5500),
//            UNIT=&UWK,SPACE=&SPASN
//PAC7RA   DD DSN=&&RA,DISP=(,PASS,DELETE),
//            DCB=(RECFM=FB,LRECL=149,BLKSIZE=11920),
//            UNIT=&UWK,SPACE=&SPASN
//PAC7RP   DD DSN=&&RP,DISP=(,PASS,DELETE),
//            DCB=(RECFM=FB,LRECL=149,BLKSIZE=23840),
//            UNIT=&UWK,SPACE=&SPASN
//PAC7DD   DD SYSOUT=&OUT
//PAC7DS   DD SYSOUT=&OUT,
//            DCB=(RECFM=FB,LRECL=67,BLKSIZE=5695)
//*
//PTU140  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU140,PTU140$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM),
//        COND=(00,NE,PTU130)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7DS   DD SYSOUT=&OUT
//PAC7NA   DD DSN=&&NA,DISP=(OLD,PASS)
//PAC7RA   DD DSN=&&RA,DISP=(OLD,PASS)
//PAC7RP   DD DSN=&&RP,DISP=(OLD,PASS)
//PAC7SR   DD DSN=&INDEXP..&ROOT.&FILE.SN,DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPASN,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7PD   DD DSN=&INDEXP..&ROOT.&FILE.SN2,DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,
//            SPACE=&SPASN,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//*
```

4.3.4. EMSN: EXTRACTION FOR SUB-NETWORK MERGE
4.3.4.1. EMSN: INTRODUCTION

EMSN: INTRODUCTION

The Extraction for Sub-Network Merge procedure (EMSN) extracts a sub-network from a database, producing a sequential file to be used as input to the Sub-Network Merge (MESN) procedure.

EXECUTION CONDITION

None, because the database is not updated directly.

Batch procedure access authorization option: Level 3 is required.

ABENDS

In case of an abend, the procedure may be restarted as it is once the problem has been corrected.

4.3.4.2. EMSN: USER INPUT

EMSN: USER INPUT

One '*' line per library to extract:

POS.	LEN.	VALUE	MEANING
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Library code
22	4	ssss	Session number (blank=current sess.)
26	1	T	Session status if Test session

Batch procedure access authorization option: The control check is made on the first '*' line.

NOTES:

The number of libraries to be extracted is limited to 99.

This set of libraries is called a 'sub-network'. The order of the extraction requests must be the same as the description of the sub-network in the Inter-library (***).

The '*' lines MUST be sorted in descending order from left to right of the sub-network; the order of the requests is not checked by the system. If even one request is invalid, all the others are also rejected.

The extracted sub-network does not need to be complete.

EXAMPLE

```
LIBRARY CODE          Corresponding extraction transactions:

AAA                          AAA is not extracted
   XXX                       (1) _*USERCODEPASSWORDXXX
      DDD                    (2) _*USERCODEPASSWORDDDD
      EEE                    (3) _*USERCODEPASSWORDEEE
         KKK                 (4) _*USERCODEPASSWORDKKK
         RRR                 (5) _*USERCODEPASSWORDRRR
   MMM                       (6) _*USERCODEPASSWORDMMM
```

PRINTED OUTPUT

The EMSN procedure prints a report stating:

- The list of applied transactions,

- The list of the sub-network libraries (including libraries which were not extracted), which corresponds to the input lines which will be required in the MESN procedure.

EXAMPLE:

ACT. CODE	LINE CODE	INITIAL LIBRARY	TARGET LIBRARY	
*	*	AAA		NOT EXTRACTED
R	*	XXX		EXTRACTED
R	*	DDD		EXTRACTED
R	*	EEE		EXTRACTED
R	*	KKK		EXTRACTED
R	*	RRR		EXTRACTED
R	*	MMM		EXTRACTED

4.3.4.3. EMSN: DESCRIPTION OF STEPS

EMSN: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

SUB-NETWORK EXTRACTION: PTU810

This step may extract up to 99 libraries.

```
.Permanent input files:
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Transaction file:
 -User input
  PAC7ME:  DSN=&&EMSNMB

.Output file:
 -Extracted sub-network
  PAC7BB:  DSN=&INDEXP..&ROOT.&FILE.&BB

.Output reports:
 -Lines required as MESN input
  PAC7EE
 -Extraction report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03


.Return codes:
 - 0:  OK.
 - 8:  Error, or no authorization on the batch procedure.
```

The return code is set when the EMSN procedure is immediately followed by the execution of the MESN procedure.

4.3.4.4. EMSN: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac     : LIBRARY COMPARISON UTILITY                           *
//*               SUB-NETWORK RETRIEVAL FOR MERGE                     *
//*********************************************************************
//$RADP.EMSN PROC FILE=$FILE,     NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//          INDEXP='$INDEXP',     NON-VSAM FILE INDEX
//          INDEX='$INDEX',       VSAM FILE INDEX
//          BB=BB,                SUFFIX OF EXTRACTED SUB-NETWORK DSN
//          SPABB='(TRK,(10,2),RLSE)',   SPACE OF EXTRACTED SUB-NETWORK
//*:        SYSTCAT='$CATV',      VSAM SYSTEM CATALOG
//*:        VSAMCAT='$CATU',      VSAM USER CATALOG
//          STEPLIB='$MODB',      LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',      SORT LIBRARY
//          UWK=$UWK,             WORK UNIT
//          OUT='$OUT',           OUTPUT CLASS
//          CYL=10,               SIZE OF SORTWORK
//          UNITS='$UNITO',       UNIT OF OUTPUT FILES
//          VOLS='SER=$VOLO',     EXTRACTED SUB-NETWORK VOLUME
//          PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//          RESLIB='$RESLIB',     IMS RESLIB
//          PROCLIB='$PRCLIB',    IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//*  INPUT  :  A USER AND LIBRARY LINE FOR EACH LIBRARY OF THE        *
//*            SUB-NETWORK TO BE EXTRACTED (25 LINES MAXIMUM)         *
//*********************************************************************
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&EMSNMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PTU810  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU810,PTU810$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
```

```
//SORTLIB  DD  DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD  UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD  UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD  UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7ME   DD DSN=&&EMSNMB,DISP=(OLD,DELETE)
//PAC7DD   DD SYSOUT=&OUT
//PAC7EE   DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7EU   DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7BB   DD DSN=&INDEXP..&ROOT.&FILE.&BB,DISP=(,CATLG),
//            VOL=&VOLS,UNIT=&UNITS,SPACE=&SPABB,
//            DCB=(RECFM=FB,LRECL=140,BLKSIZE=4760)
//*
```

4.3.5. MESN: SUB-NETWORK MERGE
4.3.5.1. MESN: INTRODUCTION

MESN: INTRODUCTION

Through the MESN procedure, one sub-network may be replaced by another sub-network extracted via the EMSN procedure.

The extracted sub-network deletes and replaces the corresponding sub-network in the Database back-up, providing a merged file which, when reorganized via REOR, will become the back-up of the new database.

THERE IS NO CONSISTENCY CHECK ON THE NEW DATABASE. THIS PROCEDURE MUST BE USED ONLY IN CASES WHERE CURRENT MANAGEMENT OF DATABASES AND SUB-NETWORKS BY THE USER ENSURES DATA CONSISTENCY.

EXECUTION CONDITION

This procedure must be preceded by the EMSN procedure, which extracts the sub-network to be merged.

The 'master' sub-network and the 'slave' sub-network must have exactly the same library hierarchy.

Batch procedure access authorization option: Level 4 is required.

ABENDS

In case of an abend, the procedure can be restarted as it is once the problem is corrected.

PRINTED OUTPUT

The procedure prints a merge report.

When input transactions do not correspond to the libraries found in the extracted sub-network, error messages are displayed, but the procedure is correctly executed.

4.3.5.2. MESN: USER INPUT

MESN : USER INPUT

Batch procedure access authorization option: One '*' line with user code and password.

One '*' line is required for each library of the sub-network, including those which are not extracted.

These lines must be coded according to the output of the EMSN procedure and, when required, with the code of the corresponding 'slave' sub-network library.

All sub-network libraries, including those which have not been extracted, must be indicated.

POS.	LEN.	VALUE	MEANING
1	1	'*'	Library not extracted
		'R'	Extracted library
2	1	'*'	Line code
3	3	aaa	'Master' sub-network library code (Required)
6	3	bbb	'Slave' sub-network library code (Default option: 'master' sub-network library code)

In case of error, the procedure is interrupted.

Example of User Input

Without code modifications:

```
**AAA
R*XXX
R*DDD
R*EEE
R*KKK
R*RRR
R*MMM
```

With code modifications:

```
**AAACEN
R*XXXAPP
R*DDD
R*EEEBIB
R*KKK
R*RRR
R*MMM
```

Although the AAA library was not extracted, the corresponding input line must be entered, with the code of the corresponding library in the target network, if it is not AAA (CEN in this example).

4.3.5.3. MESN: DESCRIPTION OF STEPS

MESN: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

SUB-NETWORK MERGE: PTU815

This step merges the sub-network extracted via the EMSN procedure with the target network.

.Permanent input files:
 -Backup file to merge
  PAC7PC: DSN=&INDEXQ..&ROOT.&FILE.PC(0)
 -Extracted sub-network
  PAC7BB: DSN=&INDEXP..&ROOT.&FILE.&BB
 -Error message file
  PAC7AE: DSN=&INDEX..&ROOT.&ROOT.AE

.Transaction file:
 -User input
  PAC7ME: DSN=&&MESNMB

.Output file:
 -Merge file to be reorganized
  PAC7CP: DSN=&NEWPC

.Output reports:
 -Merge report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

.Return code:
 .8: No authorization on the batch procedure.

The merge result MUST BE REORGANIZED (REOR procedure) before the restoration.

4.3.5.4. MESN: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : LIBRARY COMPARISON UTILITY                        *
//*                 SUB-NETWORK MERGE                                 *
//*********************************************************************
//$RADP.MESN PROC FILE=$FILE,      PHYSICAL-DATABASE NUMBER
//          ROOT=$ROOT,            ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',        VSAM FILE INDEX
//          INDEXP='$INDEXP',      NON-VSAM FILE INDEX
//          INDEXQ='$INDEXQ',      DATA GROUP FILE INDEX
//          BB=BB,                 SUFFIX OF EXTRACTED SUB-NETWORK DSN
//          NEWPC=,                MERGED BACKUP DSNAME
//          STEPLIB='$MODB',       LIBRARY OF LOAD-MODULES
//          OUT='$OUT',            OUTPUT CLASS
//          UWK=$UWK,              WORK UNIT
//          UNITS='$UNITO',        MERGED BACKUP UNIT
//          VOLS='SER=$VOLO',      MERGED BACKUP VOLUME
//          SPAPC='(TRK,(100,20),RLSE)',  SPACE OF MERGED BACKUP
//*:        SYSTCAT='$CATV',       VSAM SYSTEM CATALOG
//          PSBLIB='$PSBLIB',      LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',      LIBRARY OF DBD'S
//          RESLIB='$RESLIB',      IMS RESLIB
//          PROCLIB='$PRCLIB',     IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************************
//*  INPUT  :  A USER AND LIBRARY LINE FOR EACH LIBRARY OF THE        *
//*            SUB-NETWORK TO BE MERGED (25 LINES MAXIMUM)            *
//*********************************************************************
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&MESNMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*
//PTU815  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTU815,PTU815$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7ME   DD DSN=&&MESNMB,DISP=(OLD,DELETE)
//PAC7BB   DD DSN=&INDEXP..&ROOT.&FILE.&BB,DISP=OLD
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(0),DISP=OLD
//PAC7CP   DD DSN=&NEWPC,DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,SPACE=&SPAPC,VOL=&VOLS,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//*
```

4.4. LOAE: AE - AP RELOADING

4.4.1. LOAE: INTRODUCTION

LOAE: INTRODUCTION

The LOAE procedure restores the AE and AP indexed files when one of them (or both) is physically lost.

Restoration is performed from the last backup of the user parameters (PE file), and from the error message file (AE0).

EXECUTION CONDITION

On-line access to the AE and AP file must be closed.

ABNORMAL EXECUTIONS

Refer to Chapter "OVERVIEW", Subchapter 'ABNORMAL ENDINGS', for more details.

4.4.2. LOAE: USER INPUT

LOAE: USER INPUT

One compulsory line:

Pos.	Len.	Value	Meaning
2	6	'NRREST'	Line code

4.4.3. LOAE: DESCRIPTION OF STEPS

LOAE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

DEFINITION OF AE AND AP FILES: IDCAMS

This step executes a DELETE/DEFINE on the AE and AP files.

LOADING OF THE AE AND AP FILES: PACU80

```
.Permanent input files:
 -User parameter backup
  PAC7CE:  DSN=&INDEXQ..&ROOT.&ROOT.PE(0)
 -Initial sequential image of error messages
  PAC7LE:  DSN=&INDEXP..&ROOT.&ROOT.AE0

.Transaction file:
 -Update transactions
  PAC7MC:  DSN=&&LOAEMB

.Permanent output files:
 -Error messages
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE
 -User parameters
  PAC7AP:  DSN=&INDEX..&ROOT.&ROOT.AP

.Sort file(s):
 SORTWK01
 SORTWK02
 SORTWK03

.Output report:
 -Reconstruction report
  PAC7IJ
```

4.4.4. LOAE: EXECUTION JCL

```
//*********************************************************************
//*  VA PAC       : UPDATE OF LOCAL PARAMETERS                        *
//*********************************************************************
//$RADP.PARM PROC ROOT=$ROOT,   ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',     VSAM FILE INDEX
//*:        SYSTCAT='$CATV',    VSAM SYSTEM CATALOG
//          OUT='$OUT',         OUTPUT CLASS
//          INDEXP='$INDEXP',   INDEX OF NON-VSAM FILES
//          INDEXQ='$INDEXQ',   INDEX OF DATA GROUP FILES
//          STEPLIB='$MODB',    LIBRARY OF BATCH LOAD-MODULES
//          PSBLIB='$PSBLIB',   LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',   LIBRARY OF DBD'S
//          RESLIB='$RESLIB',   IMS RESLIB
//          PROCLIB='$PRCLIB',  IMS PROCLIB
//          SORTLIB='$BIBT',    SORT LIBRARY
//          UWK=$UWK,           WORK UNIT
//          CYL=10,             SPACE OF SORTWORKS
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&LOAEMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAP     DD DSN=&INDEX..&ROOT.&ROOT.AP,DISP=SHR
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAP),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//*
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AP),DISP=SHR
//*
//PACU80  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACU80,PACU80$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//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)
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
```

```
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AP$SUF DD DSN=&INDEX..&ROOT.&ROOT.AP,DISP=SHR
//PAC7CE   DD DSN=&INDEXQ..&ROOT.&ROOT.PE(0),DISP=(OLD,KEEP)
//PAC7IJ   DD SYSOUT=&OUT
//PAC7LE   DD DSN=&INDEXP..&ROOT.&ROOT.AE0,DISP=SHR
//PAC7MC   DD DSN=&&LOAEMB,DISP=(OLD,DELETE)
//*
```

4.5. VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY

4.5.1. VINS: INTRODUCTION

VINS: INTRODUCTION

VINS: INSTALLATION

The VINS procedure performs a batch update of the database, based on transactions provided with the product. It is used for the installation of the VA Pac/VA Smalltalk and VA Pac/TeamConnection bridges.

Entities are created in Inter-Library mode, which allows access from any Library of the network.

If some user entities have the same codes in the sub-network, VINS refuses to create them in inter-library mode, except if the update option has been set to 'F' on the '*' line. In such a case, VINS deletes all user entities with this code in the sub-network. A report then lists the user entities that have been deleted. The corresponding deletion transactions are not journalized.

EXECUTION CONDITION

On-line access must be prohibited.

Global authorization level 4 is required.

ABENDS

Refer to chapter 'OVERVIEW', sub-chapter 'Abnormal Endings'.

When the abend occurs during the execution of the PACINS program, the database is no longer consistent. Once the problem is solved, the database must be re-loaded with the retrieval of the archived transactions. The VINS procedure must then be executed again.

4.5.2. VINS: USER INPUT

VINS: INPUT-PROCESSING-RESULTS

USER INPUT

The VINS procedure requires two types of user input.

. User ID:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8		User code
11	8		Password
27	1		Update option:
			' ' - No update
			'S' - Update simulation with printing of list of U.E.'s to be cancelled
			'F' - Forcing the cancellation of U.E.'s with the same codes in lower level libraries

. Transactions used to create the necessary User Entities, which are provided on installation: the contents of these transactions MUST NOT BE MODIFIED.

PRINTED OUTPUT

The procedure prints out:

- A global report of the update,

- If the update option was set, the list of cancellation transactions.

RESULT

Once the update is performed, the network is ready for either on line or batch use.

4.5.3. VINS: DESCRIPTION OF STEPS

VINS: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

DATABASE UPDATE: PACINS

```
.Permanent update files:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN
 -Journal file
  PAC7AJ:  DSN=&INDEX..&ROOT.&FILE.AJ

.Permanent input file:
 -Error message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input-transaction files:
 -User-Entity transactions
  PAC7MV:  DSN=&FDIC
 -'*' line transaction
  PAC7MB:  DSN=&&VINSMB

.Output reports:
 -Update report
  PAC7IE (Length=132)
 -Deletion-transaction list
  PAC7EE (Length=80)
 -Batch-procedure error report
  PAC7DD
```

4.5.4. VINS: EXECUTION JCL

```
//*********************************************************************
//*  VA PAC      : BATCH UPDATE OF VA SMALLTALK-VA PAC ENTITIES       *
//*********************************************************************
//$RADP.VINS PROC FILE=$FILE,       PHYSICAL-DATABASE NUMBER
//          ROOT=$ROOT,             ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',         VSAM FILE INDEX
//          INDEXP='$INDEXP',       INDEX OF NON-VSAM FILES
//*:        VSAMCAT='$CATU',        VSAM USER CATALOG
//*:        SYSTCAT='$CATV',        VSAM SYSTEM CATALOG
//          OUT='$OUT',             OUTPUT CLASS
//          STEPLIB='$MODB',        LIBRARY OF BATCH LOAD-MODULES
//          PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//          RESLIB='$RESLIB',       IMS RESLIB
//          PROCLIB='$PRCLIB',      IMS PROCLIB
//          UWK=$UWK,               WORK UNIT
//          FDIC='$INDEXP..$ROOT.$ROOT.VINS',  DSNAME OF VA SMLTKDICT.
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM,BKO=Y
//*--------------------------------------------------------------------*
//*
//COPY    EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7MB   DD DSN=&&VINSMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//CARTE    DD DDNAME=SYSIN
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAN     DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//*
//PACINS  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PACINS,PACA15$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM,&BKO)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AJ$SUF DD DSN=&INDEX..&ROOT.&FILE.AJ,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7EE DD SYSOUT=&OUT
//PAC7EE DD SYSOUT=&OUT
//PAC7IE DD SYSOUT=&OUT
//PAC7MB DD DSN=&&VINSMB,DISP=(OLD,DELETE)
//PAC7MV DD DSN=&FDIC,DISP=SHR
//
```

4.6. RTLO: DELETION OF INVALID UPDATE LOCKS

4.6.1. RTLO: INTRODUCTION

RTLO: INTRODUCTION

The RTLO procedure deletes erroneous update locks produced by the retrieval of a previous release of the Database.

The problem is detected by the fact that an ENTITY TO BE CREATED is considered as an ENTITY LOCKED UNDER ANOTHER USER CODE. Such may be the case with Databases in which entities locked in frozen sessions have been deleted.

CHARACTERISTICS

This procedure does not entail any user input. It provides a stream of batch deletion transactions for invalid locks in the database, which is to be used as input to the Database Updating (UPDT) procedure.

EXECUTION CONDITION

On-line access must be closed.

PRINTED OUTPUT

This procedure prints out a list of the deleted invalid locks and a list of the generated batch deletion transactions.

4.6.2. RTLO: DESCRIPTION OF STEPS

RTLO: DESCRIPTION OF STEPS

RETRIEVAL OF INVALID LOCKS: PTULOI

```
.Permanent Input files:
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Permanent Input/Output files:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Index file
  PAC7AN:  DSN=&INDEX..&ROOT.&FILE.AN

.Output file:
 -Generated deletion transactions
  PAC7MB:  DSN=&&RTLOUPDT

.Output report:
 -Lists
  PAC7EU

.Internal Sort:
 SORTWK01
 SORTWK02
 SORTWK03
```

4.6.3. RTLO: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac       : LOCKED-ENTITY RETRIEVAL                           *
//*********************************************************************
//$RADP.RTLO PROC FILE=$FILE,      PHYSICAL-DATABASE NUMBER
//          ROOT=$ROOT,            ROOT OF THE VA PAC SYSTEM
//          INDEX='$INDEX',        VSAM FILE INDEX
//*:        VSAMCAT='$CATU',       VSAM USER CATALOG
//*:        SYSTCAT='$CATV',       VSAM SYSTEM CATALOG
//          OUT='$OUT',            OUTPUT CLASS
//          INDEXP='$INDEXP',      INDEX OF NON-VSAM FILES
//          SPAMV='(TRK,(30,10),RLSE)',  SPACE OF EXTRACTED TRANSACTION
//          STEPLIB='$MODB',       LIBRARY OF BATCH LOAD MODULES
//          PSBLIB='$PSBLIB',      LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',      LIBRARY OF DBD'S
//          RESLIB='$RESLIB',      IMS RESLIB
//          PROCLIB='$PRCLIB',     IMS PROCLIB
//          SORTLIB='$BIBT',       SORT LIBRARY
//          UWK=$UWK,              WORK UNIT
//          CYL=5,                 SORT-WORK SIZE
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*--------------------------------------------------------------------*
//*
//PTULOI  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,PTULOI,PTULOI$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE$SUF DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN$SUF DD DSN=&INDEX..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR$SUF DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7MB   DD DSN=&&RTLOUPDT,DISP=(,PASS),UNIT=&UWK,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//            SPACE=&SPAMV
//PAC7EU   DD SYSOUT=&OUT
//*
```

4.7. UXSR: PARTIAL SUB-NETWORK EXTRACTION

4.7.1. UXSR: INTRODUCTION

UXSR: INTRODUCTION

The Partial Sub-Network Extraction procedure (UXSR) creates a VisualAge Pacbase sub-network from an existing database, by:

. Creating Libraries (MLIB equivalent)
. Merging Libraries
. Renaming Libraries

It is also possible to select:

. A frozen session (nT):

This frozen session will become the current session in the new Database.
No other frozen session will be selected.
The image of this Database will be identical to the view which existed in the nT frozen session, but this time it will be in n+1 current session.

. The current session or all sessions (current included):

Via an option, you can select all the sessions ('T' in position 67 of the * line), or only the current session (' ' in position 67 of the * line).

EXAMPLES:

. Creation of Libraries:

```
C*CEN    AAA  (1)
C*APPCENBBB  (2)
```

(1) Creation of the CEN Library. AAA must not exist in the source Database.
(2) Creation of the APP Library in the CEN Library. BBB must not exist in the source Database.

. Merging of Libraries in the same Library:

```
C*CEN   CEN  (1)
C*APPCENAPP  (2)
C*APPCENBQQ  (2)
```

(1) Creation of the CEN Library with the contents of CEN.
(2) Creation of the APP Library under the CEN Library with the contents of APP and BQQ.

The definition of the APP Library in the new Database will be identical to that of APP in the source Database since APP comes first, before BQQ.

. Renaming of Library:

```
C*CEN   AAA  (1)
```

(1) Creation of the CEN Library with the contents of APP.

WARNING

No consistency checks are carried out; make sure you have entered valid user input lines.

EXECUTION CONDITION

On-line access must be prohibited. This procedure processes data only. It must therefore be followed by the REOR, then REST procedures, in order for the new Database to be taken into account.

4.7.2. UXSR: USER INPUT

UXSR: USER INPUT

One '*' line:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	4	nnnn	Session number (blank=current)
26	1	'T'	If selection of frozen session
		' '	If selection of current session
49	1		Option of locks extraction:
		' '	Locks extraction: user code = user code of '*' line
		'1'	No extraction of locks
		'2'	Locks extraction: user code = source user code
67	1	'T'	If col 26 = ' ' then selection of all the frozen session
		' '	If col 26 = ' ' then selection of the current session only

You must enter as many lines (optional) as Libraries to be extracted for update.

Pos.	Len.	Value	Meaning
1	1	'C'	Creation
2	1	'*'	Line code
3	3	bbb	Code of Library to be created
6	3	ccc	Code of higher Library if any
9	3	ddd	Code of source Library required even when creating a new Library, in this case enter any code not existing in the source Database.

NOTE: Do not use the character '*' in Library codes (incompatibility with the WorkStation).

4.7.3. UXSR: DESCRIPTION OF STEPS

UXSR: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

VERIFICATION OF VSAM FILES: IDCAMS

FORMATTING OF THE SEQUENTIAL IMAGE: UTIXSR

```
.Permanent input files:
 -Data file
  PAC7AR:  DSN=&INDEX..&ROOT.&FILE.AR
 -Error-message file
  PAC7AE:  DSN=&INDEX..&ROOT.&ROOT.AE

.Input transaction file:
 -Update transactions
  PAC7MB:  DSN=&&UXSRMB

.Output file:
 -Sequential image of the database
  PAC7PC:  DSN=&INDEXQ..&ROOT.&FILE.UXSR

.Output reports:
 -List of user transactions
  PAC7EV
 -Resulting Database-condition
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD
```

4.7.4. UXSR: EXECUTION JCL

```
//*******************************************************************
//* VA Pac       :  EXTRACTION FROM DATABASE FOR SUB-NETWORK CREATION *
//*******************************************************************
//$RADP.UXSR   PROC FILE=$FILE, NUMBER OF THE VA PAC SYSTEM
//         ROOT=$ROOT,          ROOT OF THE VA PAC SYSTEM
//         INDEX='$INDEX',      INDEX OF SYSTEM VSAM FILES
//         INDEXQ='$INDEXQ',    INDEX OF GENERATION FILES
//         INDEXP='$INDEXP',    INDEX OF SYSTEM NON-VSAM FILES
//*:       VSAMCAT='$CATU',     USER VSAM CATALOG
//*:       SYSTCAT='$CATV',     SYSTEM VASM CATALOG
//         OUT=$OUT,            OUTPUT CLASS
//         STEPLIB='$MODB',     LIBRARY OF BATCH LOAD MODULES
//         PSBLIB='$PSBLIB',    LIBRARY OF PSB
//         DBDLIB='$DBDLIB',    LIBRARY OF DBD
//         RESLIB='$RESLIB',    IMS RESLIB
//         PROCLIB='$PRCLIB',   IMS PROCLIB
//         UWK='$UWK',          WORK UNIT
//         VOLS='SER=$VOLO',    BACKUP VOLUME
//         UNITS=$UNITO,        BACKUP UNIT
//         SPAPC='(TRK,(100,10),RLSE)', SPACE OF THE 'PC' FILE
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//INPUT   EXEC PGM=PTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB   DD DSN=&&UXSRMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*
//VERIFY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//DDAE     DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//DDAR     DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//         DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//*
//UTIXSR  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,UTIXSR,UTIXSR$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDEXP..&ROOT.&ROOT.SY(DFSVSAM8),DISP=SHR
//PAC7AE$SUF   DD DSN=&INDEX..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR$SUF   DD DSN=&INDEX..&ROOT.&FILE.AR,DISP=SHR
//PAC7MB       DD DSN=&&UXSRMB,DISP=(OLD,DELETE)
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),
```

```
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,
//            VOL=&VOLS,
//            SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//PAC7EV   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

5. MIGRATIONS

5.1. CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS

5.1.1. CRYP: INTRODUCTION

CRYP: INTRODUCTION

The CRYP procedure performs the encryption and decryption of user passwords in the PE user-parameter backup file.

The objective of this procedure is to transfer the PE file onto platforms with different codings.

EXECUTION CONDITION

Authorization level '4' for the update of user parameters (PARM).

5.1.2. CRYP: USER INPUT

CRYP: USER INPUT

A '*' line with the user code and the password must be entered.

The user code specified on the '*' line must exist in the PE file to be processed.

The procedure's specific user input allows for the selection of either Encryption or Decryption.

Pos.	Len.	Value	Meaning
3	6	'CODE'	Password encryption
		'DECODE'	Password decryption

NOTE: When decrypting, the backup obtained must not be reloaded via the 'PARM' procedure. If it were, user passwords would no longer be recognized.

5.1.3. CRYP: DESCRIPTION OF STEPS

CRYP : DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

ENCRYPTION / DECRYPTION OF PASSWORDS: PACU99

```
.Input files:
 -User parameter backup
  PAC7CE:  DSN=&INFILE
 -User input
  PAC7MB:  DSN=&&CRYPMB

.Output file:
 -User parameter backup
  PAC7EC:  DSN=&OUTFILE

.Output report:
 -Execution report
  PAC7DD
```

5.1.4. CRYP: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac  : ENCRYPTION-DECRYPTION OF USER PASSWORDS FROM THE USER *
//*            PARAMETERS BACKUP                                     *
//*********************************************************************
//$RADP.CRYP  PROC OUT=X,            OUTPUT CLASS
//         STEPLIB='$MODB',          LIBRARY OF BATCH LOAD-MODULES
//         INFILE=,                  INPUT FILE DSNAME
//         OUTFILE=,                 OUTPUT FILE DSNAME
//         VOLS='$VOLO',             OUTPUT FILE VOLUME
//         UNITS='$UNITO',           OUTPUT FILE UNIT (DISK OR TAPE)
//         UWK='$UWK',               WORK UNIT
//         SPACY='(TRK,(15,1),RLSE)' OUTPUT FILE SPACE
//*********************************************************************
//*   FORMAT OF TRANSACTIONS AS INPUT :                               *
//*   .. A  * USER CODE AND PASSWORD LINE                             *
//*      (AUTORIZATION LEVEL 4 ON PARAMETERS                          *
//*   .. A COMMAND LINE                                               *
//*      COL  3-8  : 'CODE'   PASSWORD ENCRYPTION                     *
//*                : 'DECODE' PASSWORD DECRYPTION                     *
//*********************************************************************
//INPUT  EXEC PGM=PTU001
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&CRYPMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,1),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PACU99 EXEC PGM=PACU99
//**********************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&CRYPMB,DISP=(OLD,PASS)
//PAC7CE   DD DSN=&INFILE,DISP=SHR
//PAC7EC   DD DSN=&OUTFILE,DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,SPACE=&SPACY,VOL=SER=&VOLS,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT
//*********************************************************************
```

5.2. LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE

5.2.1. LVBL: INTRODUCTION

LVBL: INTRODUCTION

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

The purpose of this procedure is to transfer the PC file onto different platforms while avoiding problems due to the presence of low-values at the time of transfer.

UTILIZATION OPTION

The LVBL procedure allows you to keep only records of the 'data' type. See the 'Description of Steps' section for further details on the implementation of this option.

EXECUTION CONDITION

None

5.2.2. LVBL: DESCRIPTION OF STEPS

LVBL: DESCRIPTION OF STEPS

REPLACEMENT OF LOW-VALUES WITH BLANKS: PTULVB

.EXEC line:

Specify PARM=DATA in order to keep only the
'data'-type records in the output file.

To keep both 'index' and 'data' records, do not specify
anything.

.Input file:
-Database backup
PAC7MC: DSN=&INDEXQ..&ROOT.&FILE.PC(0)

.Output file:
-Database backup
PAC7PC: DSN=&INDEXQ..&ROOT.&FILE.PC(+1)

5.2.3. LVBL: EXECUTION JCL

```
//*********************************************************************
//*  VA Pac   : REPLACEMENT OF LOW-VALUES WITH BLANKS IN PC FILE      *
//*  (OPTION: SUBMIT PROCEDURE WITH PARM=(DATA) TO PROCESS DATA ONLY) *
//*********************************************************************
//$RADP.LVBL PROC FILE=$FILE,  NUMBER OF THE PHYSICAL DATABASE
//         ROOT=$ROOT,         ROOT OF THE VA PAC SYSTEM
//         INDEXQ='$INDEXQ',   DATA GROUP FILE INDEX
//         STEPLIB='$MODB',    LIBRARY OF BATCH LOAD-MODULES
//         OUT=X,              OUTPUT CLASS
//         UNITS='$UNITO',     BACKUP UNIT
//         SPAPC='(TRK,(020,10),RLSE) BACKUP SPACE
//*********************************************************************
//PTULVB EXEC PGM=PTULVB,PARM=' '
//*****************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7MC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(0),DISP=SHR
//PAC7PC   DD DSN=&INDEXQ..&ROOT.&FILE.PC(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,SPACE=&SPAPC,
//            DCB=&INDEXQ..DSCB.&ROOT.&FILE.PC
//SYSUDUMP DD SYSOUT=&OUT
//*
```

5.3. SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION

5.3.1. SMTD: INTRODUCTION

SMTD: INTRODUCTION

The SMTD procedure backs up the TD table-description file by transforming binary characters into their display format.

The aim of the procedure is to transfer the TD file onto different platforms while avoiding problems caused by the presence of these characters at the time of transfers.

EXECUTION CONDITION

None.

USER INPUT

None.

5.3.2. SMTD: DESCRIPTION OF STEPS

SMTD: DESCRIPTION OF STEPS

VERIFICATION OF DT FILE: IDCAMS

BACKUP OF TD FILE: PTATDM

```
.Permanent input file:
 -Table descriptions file:
  PAC7TD:  DSN=&INDEX..&ROOT.&FILE.DT

.Output file:
 -BACKUP OF TABLE DESCRIPTIONS
  FOR MIGRATION
  PAC7TC:  DSN=&INDEXP..&ROOT.SMTD
```

5.3.3. SMTD: EXECUTION JCL

```
//*********************************************************************
//*  Pactables     :  BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION      *
//*********************************************************************
//$RADP.SMTD PROC FILE=$FILE,     NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//          INDEXP='$INDEXP',     NON-VSAM FILE INDEX
//          INDEX='$INDEX',       VSAM FILE INDEX
//*:        VSAMCAT='$CATU',      VSAM USER CATALOG
//          STEPLIB='$MODB',      LIBRARY OF BATCH LOAD-MODULES
//          OUT='$OUT',           OUTPUT CLASS
//          UNITS='$UNITO',       UNIT
//          VOLS='SER=$VOLP',     VOLUME
//          SPATD='(TRK,(15,5),RLSE)' 'TD' BACKUP SPACE
//*--------------------------------------------------------------------*
//*
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//DDDTN    DD DSN=&INDEX..&ROOT.&FILE.DT,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(VERIFDT),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//PTATDM  EXEC PGM=PTATDM
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7TC   DD DSN=&INDEXP..&ROOT.SMTD,
//            UNIT=&UNITS,VOL=&VOLS,
//            DISP=(,CATLG,DELETE),SPACE=&SPATD,
//            DCB=(RECFM=FB,LRECL=244,BLKSIZE=2440)
//PAC7TD   DD DSN=&INDEX..&ROOT.&FILE.DT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
===SEQ FOR PEI
```

5.4. RMTD: RESTORATION OF TABLE DESCRIPTIONS

5.4.1. RMTD: INTRODUCTION

RMTD: INTRODUCTION

The Restoration of Table Descriptions procedure (RMTD) restores the TD file of Table Descriptions from its TC sequential backup produced by the SMTD procedure.

This procedure entails no execution condition and no user input.

5.4.2. RMTD: DESCRIPTION OF STEPS

RMTD: DESCRIPTION OF STEPS

DEFINITION OF DT FILE: IDCAMS

This step performs a DELETE/DEFINE of DT file.

RESTORATION OF TD FILE: PTATDR

```
.Backup input file:
 -Sequential file of table descriptions
  PAC7TC:  DSN=&INDEXP..&ROOT.SMTD

.Output file:
 -Table descriptions file
  PAC7TD:  DSN=&INDEX..&ROOT.&FILE.DT
```

5.4.3. RMTD: EXECUTION JCL

```
//*********************************************************************
//*  PACTABLES        TABLES BACKUP AFTER MIGRATION                   *
//*********************************************************************
//$RADP.RMTD PROC FILE=$FILE,     NUMBER OF THE PHYSICAL DATABASE
//          ROOT=$ROOT,           ROOT OF THE VA PAC SYSTEM
//          INDEXP='$INDEXP',     NON-VSAM FILE INDEX
//          INDEX='$INDEX',       VSAM FILE INDEX
//*:        VSAMCAT='$CATU',      VSAM USER CATALOG
//          STEPLIB='$MODB',      LIBRARY OF BATCH LOAD-MODULES
//          OUT='$OUT'            OUTPUT CLASS
//*--------------------------------------------------------------------*
//*
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSIN    DD DSN=&INDEXP..&ROOT.&ROOT.SY(DF&ROOT.&FILE.DT),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//PTATDR  EXEC PGM=PTATDR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7TC   DD DSN=&INDEXP..&ROOT.SMTD,DISP=SHR
//PAC7TD   DD DSN=&INDEX..&ROOT.&FILE.DT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
```

5.5. RPTD: TABLE DESCRIPTIONS RETRIEVAL

5.5.1. RPTD: INTRODUCTION

RPTD: INTRODUCTION

The RPTD procedure must be used to retrieve the TD backup file from a previous release, so as to make it usable by the RMTD, Rel. 2.0, restoration procedure.

RPTD adds the century mark to all dates used in table-descriptions handling. The pivot year for century change must be parameterized.

EXECUTION CONDITION

None.

PRINTOUT

The RPTD procedure prints a report on the retrieval.

5.5.2. RPTD: USER INPUT

USER INPUT

.One parameter line defining the pivot year for adding
the century mark.

Pos.	Len.	Value	Meaning
1	2	2 digits other than '00'	Pivot Year

5.5.3. RPTD: DESCRIPTION OF STEPS

RPTD : DESCRIPTION OF STEPS

2.0 RETRIEVAL OF TD FILE: PTAR20

```
.Input files:
 -Table-descriptions backup
  PAC7MB  :    DSN=&&RPTDMB
 -User parameter-line
  PAC7TC  :    DSN=&OLDTDSAV

.Output file:
 -2.0 backup of table-descriptions
  PAC7TR  :    DSN=&TDSAV

.Output report:
 -Retrieval report
  PAC7ET
```

5.5.4. RPTD: EXECUTION JCL

```
//*************************************************************
//*          - RETRIEVAL 2.0 OF TD FILE FOR MIGRATION -
//*************************************************************
//$RADP.RPTD PROC TDSAV=,    DSN OF 2.0 TABLES DESCRIPTION BACKUP
//         OLDTDSAV=,       DSN OF <2.0 TABLES DESCRIPTION BACKUP
//         STEPLIB='$MODB',               LIBRARY OF LOAD-MODULES
//*:        VSAMCAT='$CATU',                    VSAM USER CATALOG
//         OUT=$OUT,                                OUTPUT CLASS
//         VOLS='SER=$VOLU',                      BACKUP VOLUME
//         UNITS=$UNITU,          BACKUP UNIT (DISK OR CARTRIDGE)
//         UWK=$UWK,                                   WORK UNIT
//         SPATD='(TRK,(150,10))'                   BACKUP SPACE
//*************************************************************
//INPUT    EXEC PGM=PTU001
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&RPTDMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PTAR20   EXEC PGM=PTAR20
//************************
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7MB   DD DSN=&&RPTDMB,DISP=(OLD,DELETE)
//PAC7TC   DD DSN=&OLDTDSAV,
//         DISP=SHR
//PAC7TR   DD DSN=&TDSAV,
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         DISP=(,CATLG,DELETE),SPACE=&SPATD,
//         DCB=(RECFM=VB,LRECL=1067,BLKSIZE=10674)
//PAC7ET   DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```