



VisualAge Pacbase 2.5

**VA PAC 2.5 : IBM CICS/OS/MVS
OPERATIONS MANUAL VOLUME II : ADMINISTRATOR'S GUIDE**

DEPCI002251A

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 (April 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.

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

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

NOTICES

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

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

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

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

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

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

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

TRADEMARKS

IBM is a trademark of International Business Machines Corporation, Inc. AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, PACBASE, RACF, RS/6000, SQL/DS, TeamConnection, and VisualAge are trademarks of International Business Machines Corporation, Inc. in the United States and/or other countries.

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

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

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

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

TABLE OF CONTENTS

1. OVERVIEW.....	9
1.1. USER IDENTIFICATION (*).....	12
1.2. ACCESS RIGHTS: BATCH-PROCEDURE AUTHOR. OPTION	14
1.3. ABNORMAL ENDINGS	17
2. DATABASE MANAGEMENT UTILITIES	18
2.1. MLIB: DATABASE MANAGEMENT	19
2.1.1. MLIB: INTRODUCTION.....	19
2.1.2. MLIB: INPUT - PROCESSING - RESULTS.....	20
2.1.3. MLIB: DESCRIPTION OF STEPS.....	24
2.1.4. MLIB: EXECUTION JCL.....	26
2.2. SAVE: DATABASE BACKUP	28
2.2.1. SAVE: INTRODUCTION.....	28
2.2.2. SAVE: PROCESSING - RESULTS	30
2.2.3. SAVE: DESCRIPTION OF STEPS.....	32
2.2.4. SAVE: EXECUTION JCL.....	34
2.3. SASY: DATABASE SYSTEM BACKUP COMPLEMENT	36
2.3.1. SASY: INTRODUCTION.....	36
2.3.2. SASY: DESCRIPTION OF STEPS.....	38
2.3.3. SASY: EXECUTION JCL.....	39
2.4. REST: DATABASE RESTORATION.....	40
2.4.1. REST: INTRODUCTION.....	40
2.4.2. REST: USER INPUT	41
2.4.3. REST: DESCRIPTION OF STEPS	45
2.4.4. REST: EXECUTION JCL	51
2.5. RESY: DATABASE SYSTEM RESTORATION COMPLEMENT	53
2.5.1. RESY: INTRODUCTION.....	53
2.5.2. RESY: USER INPUT - RESULTS.....	55
2.5.3. RESY: DESCRIPTION OF STEPS	58
2.5.4. RESY: EXECUTION JCL.....	62
2.6. ARCH: JOURNAL ARCHIVAL.....	64
2.6.1. ARCH: INTRODUCTION.....	64
2.6.2. ARCH: INPUT - RECOMMENDATIONS - RESULTS.....	65
2.6.3. ARCH: DESCRIPTION OF STEPS.....	68
2.6.4. ARCH: EXECUTION JCL.....	71
2.7. REOR: DATABASE REORGANIZATION	73
2.7.1. REOR: INTRODUCTION.....	73
2.7.2. REOR: INPUT - RECOMMENDATIONS	75
2.7.3. REOR: DESCRIPTION OF STEPS.....	78
2.7.4. REOR: EXECUTION JCL.....	83
2.8. SVAG: GENERATION-PRINT REQUEST BACKUP.....	86
2.8.1. SVAG: INTRODUCTION.....	86
2.8.2. SVAG: DESCRIPTION OF STEPS	87
2.8.3. SVAG: EXECUTION JCL.....	88
2.9. REAG: GENERATION-PRINT REQUEST RESTORATION	89
2.9.1. REAG: INTRODUCTION.....	89
2.9.2. REAG: USER INPUT	90
2.9.3. REAG: DESCRIPTION OF STEPS.....	91
2.9.4. REAG: EXECUTION JCL.....	93
2.10. PARM: UPDATE OF USER PARAMETERS.....	94
2.10.1. PARM: INTRODUCTION.....	94
2.10.2. PARM: INPUT - RECOMMENDATIONS	97
2.10.3. PARM: USER-CODE DEFINITION	104
2.10.4. PARM: USER-CODE GLOBAL AUTHORIZATIONS.....	107
2.10.5. PARM: USER-CODE SPECIFIC AUTHORIZATIONS	110
2.10.6. PARM: TEXT TYPES.....	113
2.10.7. PARM: MODIFICATION OF STANDARD ERROR MESSAGES.....	115
2.10.8. PARM: GENERATED-STREAM CONTROL CARDS.....	118

2.10.9. PARM: DESCRIPTION OF STEPS.....	126
2.10.10. PARM: EXECUTION JCL.....	129
3. VERSIONING UTILITIES	131
3.1. PEI: PRODUCTION ENVIRONMENT INTERFACE	132
3.1.1. PEI: OVERVIEW.....	132
3.1.2. INPE: FILE INITIALIZATION.....	134
3.1.2.1. INPE: INTRODUCTION.....	134
3.1.2.2. INPE: DESCRIPTION OF STEPS	135
3.1.2.3. INPE: EXECUTION JCL	136
3.1.3. SVPE: FILE BACKUP.....	137
3.1.3.1. SVPE: INTRODUCTION.....	137
3.1.3.2. SVPE: DESCRIPTION OF STEPS	138
3.1.3.3. SVPE: EXECUTION JCL	140
3.1.4. RSPE: FILE RESTORATION	141
3.1.4.1. RSPE: INTRODUCTION	141
3.1.4.2. RSPE: DESCRIPTION OF STEPS.....	142
3.1.4.3. RSPE: EXECUTION JCL.....	144
3.1.5. PRPE: PRODUCTION ENVIRONMENT PRINTOUTS.....	145
3.1.5.1. PRPE: INTRODUCTION	145
3.1.5.2. PRPE: USER INPUT.....	146
3.1.5.3. PRPE: DESCRIPTION OF STEPS	147
3.1.5.4. PRPE: EXECUTION JCL.....	148
3.1.6. GRPE: TRANSACTION-GENERATION FOR REORGANIZATION.....	149
3.1.6.1. GRPE: INTRODUCTION	149
3.1.6.2. GRPE: DESCRIPTION OF STEPS	150
3.1.6.3. GRPE: EXECUTION JCL	151
3.1.7. HIPE: AUTOMATIC SESSION FREEZE.....	152
3.1.7.1. HIPE: INTRODUCTION.....	152
3.1.7.2. HIPE: USER INPUT.....	153
3.1.7.3. HIPE: DESCRIPTION OF STEPS	154
3.1.7.4. HIPE: EXECUTION JCL	156
3.1.8. SIPE: PRODUCTION TURNOVER SIMULATION	157
3.1.8.1. SIPE: INTRODUCTION	157
3.1.8.2. SIPE: USER INPUT	158
3.1.8.3. SIPE: DESCRIPTION OF STEPS.....	159
3.1.8.4. SIPE: EXECUTION JCL.....	160
3.2. PAC/TRANSFER.....	161
3.2.1. TRUP: TRANSFER-PARAMETER UPDATE.....	163
3.2.1.1. TRUP: INTRODUCTION	163
3.2.1.2. TRUP: USER INPUT	166
3.2.1.3. TRUP: DESCRIPTION OF STEPS	172
3.2.1.4. TRUP: EXECUTION JCL	174
3.2.2. TRJC: COMPRESSION OF ARCHIVED JOURNAL.....	176
3.2.2.1. TRJC: INTRODUCTION	176
3.2.2.2. TRJC: USER INPUT	178
3.2.2.3. TRJC: DESCRIPTION OF STEPS.....	179
3.2.2.4. TRJC: EXECUTION JCL.....	181
3.2.3. TRPF: TRANSFER-FILE CREATION.....	183
3.2.3.1. TRPF: INTRODUCTION	183
3.2.3.2. TRPF: USER INPUT.....	184
3.2.3.3. TRPF: DESCRIPTION OF STEPS.....	185
3.2.3.4. TRPF: EXECUTION JCL.....	186
3.2.4. TRDU: DSMS-ENVIRONMENT PREPARATION.....	187
3.2.4.1. TRDU: INTRODUCTION.....	187
3.2.4.2. TRDU: USER INPUT.....	190
3.2.4.3. TRDU: DESCRIPTION OF STEPS	191
3.2.4.4. TRDU: EXECUTION JCL	194
3.2.5. UPDATE OF DSMS FUNCTION BEFORE VA PAC UPDATE	196
3.2.6. TRRP: GENERATION OF TRANSFER TRANSACTIONS.....	197
3.2.6.1. TRRP: INTRODUCTION.....	197
3.2.6.2. TRRP: USER INPUT.....	200
3.2.6.3. TRRP: DESCRIPTION OF STEPS	201
3.2.6.4. TRRP: EXECUTION JCL	204
3.2.7. UPDATE OF THE VISUALAGE PACBASE DATABASE.....	206
3.2.8. REINITIALIZATION OF THE DSMS ENVIRONMENT.....	207

3.3. TEAMCONNECTION	208
3.3.1. TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION	209
3.3.1.1. TCGP: INTRODUCTION.....	209
3.3.1.2. TCGP: DESCRIPTION OF STEPS	210
3.3.1.3. TCGP: EXECUTION JCL	211
3.3.2. TCCI: INTER-ENVIRONMENT INTEGRITY CHECK.....	213
3.3.2.1. TRCI: INTRODUCTION.....	213
3.3.2.2. TRCI: DESCRIPTION OF STEPS	214
3.3.2.3. TCCI: EXECUTION.....	216
3.3.3. TCLS: LIBRARY-SESSION UPDATE	218
3.3.3.1. TCLS: INTRODUCTION	218
3.3.3.2. TCLS: USER INPUT.....	219
3.3.3.3. TCLS: DESCRIPTION OF STEPS	221
3.3.3.4. TCLS: EXECUTION JCL.....	222
4. MANAGER'S UTILITIES.....	223
4.1. STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES	224
4.1.1. STOP: INTRODUCTION	224
4.1.2. STOP: USER INPUT.....	225
4.1.3. STOP: DESCRIPTION OF STEPS.....	226
4.1.4. STOP: EXECUTION JCL.....	227
4.2. SESSION MANAGEMENT	228
4.2.1. ESES - CSES: INTRODUCTION.....	228
4.2.2. ESES: EXTRACTION OF SESSION NUMBERS.....	229
4.2.3. ESES: DESCRIPTION OF STEPS	231
4.2.4. ESES: EXECUTION JCL	232
4.2.5. CSES: COMPRESSION OF SESSION NUMBERS.....	233
4.2.6. CSES: USER INPUT	234
4.2.7. CSES: DESCRIPTION OF STEPS	235
4.2.8. CSES: EXECUTION JCL.....	237
4.3. GBIR: PARTITIONED DATABASE MANAGER	239
4.3.1. GBIR: INTRODUCTION.....	239
4.3.2. CPSN: SUB-NETWORK COMPARISON.....	242
4.3.2.1. CPSN: INTRODUCTION.....	242
4.3.2.2. CPSN: NOTES ON THE RESULTS	243
4.3.2.3. CPSN: DESCRIPTION OF STEPS	244
4.3.2.4. CPSN: EXECUTION JCL	246
4.3.3. SASN: SUB-NETWORK BACKUP	247
4.3.3.1. SASN: INTRODUCTION.....	247
4.3.3.2. SASN: USER INPUT	248
4.3.3.3. SASN: DESCRIPTION OF STEPS	249
4.3.3.4. SASN: EXECUTION JCL.....	251
4.3.4. EMSN: EXTRACTION FOR SUB-NETWORK MERGE.....	253
4.3.4.1. EMSN: INTRODUCTION.....	253
4.3.4.2. EMSN: USER INPUT.....	254
4.3.4.3. EMSN: DESCRIPTION OF STEPS	256
4.3.4.4. EMSN: EXECUTION JCL.....	258
4.3.5. MESN: SUB-NETWORK MERGE.....	259
4.3.5.1. MESN: INTRODUCTION.....	259
4.3.5.2. MESN: USER INPUT.....	260
4.3.5.3. MESN: DESCRIPTION OF STEPS	262
4.3.5.4. MESN: EXECUTION JCL.....	263
4.4. LOAE: AE - AP RELOADING	264
4.4.1. LOAE: INTRODUCTION.....	264
4.4.2. LOAE: USER INPUT.....	265
4.4.3. LOAE: DESCRIPTION OF STEPS	266
4.4.4. LOAE: EXECUTION JCL	267
4.5. VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY	268
4.5.1. VINS: INTRODUCTION	268
4.5.2. VINS: USER INPUT.....	269
4.5.3. VINS: DESCRIPTION OF STEPS.....	270
4.5.4. VINS: EXECUTION JCL.....	271
4.6. RTLO: DELETION OF INVALID UPDATE LOCKS	272
4.6.1. RTLO: INTRODUCTION	272

4.6.2. RTLO: DESCRIPTION OF STEPS.....	273
4.6.3. RTLO: EXECUTION JCL.....	274
4.7. UXSR: PARTIAL SUB-NETWORK EXTRACTION	275
4.7.1. UXSR: INTRODUCTION.....	275
4.7.2. UXSR: USER INPUT.....	277
4.7.3. UXSR: DESCRIPTION OF STEPS	278
4.7.4. UXSR: EXECUTION JCL.....	279
5. MIGRATIONS.....	280
5.1. CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS	281
5.1.1. CRYP: INTRODUCTION	281
5.1.2. CRYP: USER INPUT.....	282
5.1.3. CRYP: DESCRIPTION OF STEPS.....	283
5.1.4. CRYP: EXECUTION JCL.....	284
5.2. LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE	285
5.2.1. LVBL: INTRODUCTION.....	285
5.2.2. LVBL: DESCRIPTION OF STEPS.....	286
5.2.3. LVBL: EXECUTION JCL.....	287
5.3. SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION.....	288
5.3.1. SMTD: INTRODUCTION.....	288
5.3.2. SMTD: DESCRIPTION OF STEPS.....	289
5.3.3. SMTD: EXECUTION JCL.....	290
5.4. RMTD: RESTORATION OF TABLE DESCRIPTIONS	291
5.4.1. RMTD: INTRODUCTION.....	291
5.4.2. RMTD: DESCRIPTION OF STEPS	292
5.4.3. RMTD: EXECUTION JCL.....	293
5.5. RPTD: TABLE DESCRIPTIONS RETRIEVAL.....	294
5.5.1. RPTD: INTRODUCTION.....	294
5.5.2. RPTD: USER INPUT.....	295
5.5.3. RPTD: DESCRIPTION OF STEPS	296
5.5.4. RPTD: EXECUTION JCL	297

VisualAge Pacbase - Operation Manual
BATCH PROC.: ADMINISTRATOR'S GUIDE
OVERVIEW

PAGE 9

1

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 Pactransfer)
- 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	!

OVERVIEW

1

USER IDENTIFICATION (*)

1

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 procedures, 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

OVERVIEW

1

ACCESS RIGHTS: BATCH-PROCEDURE AUTHOR. OPTION

2

! PROCEDURE !	GLOBAL !	DATABASE !
! !	AUTHORIZATION !	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

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

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

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

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

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

! NN !	! STATUS	!	! OO !	! OPERATION	!
! 21 !	! SEQUENCE ERROR	!	! !	!	!
! 22 !	! DUPLICATE KEY	!	! W !	! WRITE	!
! 23 !	! NO RECORD FOUND	!	! RW !	! REWRITE	!
! 24 !	! BOUNDARY VIOLATION (KSDS-RRDS)!	!	! RU !	! READ UP	!
! 30 !	! SYSTEM ERROR	!	! OP !	! OPEN	!
! 34 !	! BOUNDARY VIOLATION(SEQUENTIAL)!	!	! CL !	! CLOSE	!
! 92 !	! LOGIC ERROR (FOR EXAMPLE, ! OPEN OF AN ALREADY OPEN FILE)	!	! D !	! DELETE	!
!	!	!	! R !	! READ	!
! 93 !	! FILE STILL OPEN UNDER CICS	!	! P !	! START	!
! 95 !	! INVALID OR INCOMPLETE FILE	!	! RN !	! READ NEXT	!
!	! INFORMATION	!	! !	!	!

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

VisualAge Pacbase - Operation Manual
BATCH PROC.: ADMINISTRATOR'S GUIDE
DATABASE MANAGEMENT UTILITIES

PAGE 18

2

2. DATABASE MANAGEMENT UTILITIES

	PAGE	19
DATABASE MANAGEMENT UTILITIES		2
MLIB: DATABASE MANAGEMENT		1
MLIB: INTRODUCTION		1

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.

DATABASE MANAGEMENT UTILITIES
MLIB: DATABASE MANAGEMENT
MLIB: INPUT - PROCESSING - RESULTS

PAGE

21

2
1
2

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.

	PAGE	22
DATABASE MANAGEMENT UTILITIES		2
MLIB: DATABASE MANAGEMENT		1
MLIB: INPUT - PROCESSING - RESULTS		2

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.

	PAGE	23
DATABASE MANAGEMENT UTILITIES		2
MLIB: DATABASE MANAGEMENT		1
MLIB: INPUT - PROCESSING - RESULTS		2

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=&INDUV..&ROOT.&FILE.AR
- Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
- Printing-generation request file
(in input-output if no simulation)
PAC7AG: DSN=&INDUV..&ROOT.&FILE.AG
- Error message file
(in input-output)
PAC7AE: DSN=&INDSV..&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

DATABASE MANAGEMENT UTILITIES
MLIB: DATABASE MANAGEMENT
MLIB: DESCRIPTION OF STEPS

2
1
3

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=&INDUV..&ROOT.&FILE.AR

.Temporary files:

-The 4 output files from the preceding step.

.Output file:

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

.Output reports:

-None.

DATABASE MANAGEMENT UTILITIES

2

MLIB: DATABASE MANAGEMENT

1

MLIB: EXECUTION JCL

4

2.1.4. MLIB: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - LIBRARIAN - *
//*****
// $RADP.MLIB PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDSN='$INDSN', INDEX OF SYSTEM NON VSAM FILES
// INDUV='$INDUV', INDEX OF USER VSAM FILES
// INDUN='$INDUN', INDEX OF USER NON VSAM FILES
// *: VSAMCAT='$VCAT', USER VSAM CATALOG
// *: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// DSCB='$DSCB', DSCB MODEL FILE
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', DATABASE BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// VOLS2='SER=$VOLUN', DATABASE 2ND BACKUP VOLUME
// UNITS2=$UNITUN, 2ND BACKUP UNIT (DISK OR CARTRIDGE)
// UWK=$UWK, WORK UNIT
// SORTLIB='$BIBT', SORT LIBRARY
// CYL=1, SORTWORK SPACE
// SPAAN='(TRK,(50,10),RLSE)', TEMPORARY INDEX SPACE
// SPAAR='(TRK,(50,10),RLSE)', TEMPORARY DATA SPACE
// SPAPC='(TRK,(100,10),RLSE)', MODIFIED VA PAC DATABASE SPACE
// SPAPD='(TRK,(100,10),RLSE)', MODIFIED VA PAC DATABASE 2
//*****
// *
// * INPUTS : *
// * * *
// * FIRST LINE (REQUIRED): *
// * COL 2 : 'G' *
// * COL 3 : 'I' TO INITIALIZE A DATABASE, OTHERWISE BLANK. *
// * COL 4 : 'S' TO SIMULATE MODIFICATIONS TO THE DATABASE, *
// * BLANK FOR COMPLETE EXECUTION *
// * * *
// * DETAIL LINE (ONE PER MODIFICATION TO THE DATABASE): *
// * COL 1 : TRANSACTION CODE ( C=CREATION, M=MODIF, D=DELETE) *
// * COL 2 : '*' *
// * COL 3 TO 5 : CODE OF THE LIBRARY TO BE CREATED OR *
// * CODE OF THE LIBRARY TO DELETE OR *
// * CODE OF THE LIBRARY WHOSE CENTRAL LIBRARY *
// * IS TO BE MODIFIED *
// * COL 6 TO 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=3440)
// VERIFY EXEC PGM=IDCAMS
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// SYSPRINT DD SYSOUT=&OUT
// PAC7AG DD DSN=&INDUV..&ROOT.&FILE.AG,DISP=SHR
// PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
// PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
// SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAG),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
// PTU100 EXEC PGM=PTU100
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR

```

DATABASE MANAGEMENT UTILITIES

2

MLIB: DATABASE MANAGEMENT

1

MLIB: EXECUTION JCL

4

```

//PAC7AG DD DSN=&INDUV..&ROOT.&FILE.AG,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR,
// AMP=( 'BUFNI=1, BUFND=3' )
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR,
// AMP=( 'BUFND=3' )
//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)
//SYSOUT DD SYSOUT=&OUT
//PAC7EU DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7EV DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PTU120 EXEC PGM=PTU120,COND=(0,NE,PTU100),REGION=0K
//*****
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//PAC7AN DD DSN=&&MLIBAN,DISP=(OLD,DELETE)
//PAC7NB DD DSN=&&MLIBNB,DISP=(OLD,DELETE)
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7PR DD DSN=&&MLIBPR,DISP=(OLD,DELETE)
//PAC7PQ DD DSN=&&MLIBPQ,DISP=(OLD,DELETE)
//PAC7PC DD DSN=&INDUN..&ROOT.&FILE.PC(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPC,
// DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7PD DD DSN=&INDUN..&ROOT.&FILE.PD(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS2,
// VOL=&VOLS2,
// SPACE=&SPAPD,
// DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//SYSUDUMP DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

DATABASE MANAGEMENT UTILITIES	PAGE	28
SAVE: DATABASE BACKUP		2
SAVE: INTRODUCTION		1

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.

	PAGE	29
DATABASE MANAGEMENT UTILITIES		2
SAVE: DATABASE BACKUP		2
SAVE: INTRODUCTION		1

ARCHIVAL AND BACKUP LINKING

If the backup procedure is preceded by a Journal archival (ARCH procedure), its execution may be conditioned 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.)

DATABASE MANAGEMENT UTILITIES
SAVE: DATABASE BACKUP
SAVE: PROCESSING - RESULTS

PAGE

30

2
2
2

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 !  
-----
```

DATABASE MANAGEMENT UTILITIES	PAGE	31
SAVE: DATABASE BACKUP		2
SAVE: PROCESSING - RESULTS		2

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.)

DATABASE MANAGEMENT UTILITIES
SAVE: DATABASE BACKUP
SAVE: DESCRIPTION OF STEPS

PAGE

32

2
2
3

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=&INDUV..&ROOT.&FILE.AR
-Error message file
PAC7AE: DSN=&INDSV..&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=&INDUV..&ROOT.&FILE.AR

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

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

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

	PAGE	33
DATABASE MANAGEMENT UTILITIES	2	
SAVE: DATABASE BACKUP	2	
SAVE: DESCRIPTION OF STEPS	3	

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=&INDUN.&ROOT.&FILE.PC(+1)
If backup Dispatch option:
-Database sequential image í2
  DDPD: DSN=&INDUN.&ROOT.&FILE.PD(+1)
```

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

DATABASE MANAGEMENT UTILITIES

SAVE: DATABASE BACKUP

SAVE: EXECUTION JCL

2

2

4

2.2.4. SAVE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - BACKUP OF THE DATABASE - *
//*****
// $RADP.SAVE PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSN='$INDSN', INDEX OF SYSTEM NON VSAM FILES
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDUV='$INDUV', INDEX OF USER VSAM FILES
// INDUN='$INDUN', INDEX OF USER NON VSAM FILES
// *: VSAMCAT='$VCAT', USER VSAM CATALOG
// *: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// UWK=$UWK, WORK UNIT
// DSCB='$DSCB', DSCB MODEL FILE
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', DATABASE BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// VOLS2='SER=$VOLUN', DATABASE 2ND BACKUP VOLUME
// UNITS2=$UNITUN, 2ND BACKUP UNIT (DISK OR CARTRIDGE)
// SPAPC='(TRK,(100,10),RLSE)', VA PAC DATABASE BACKUP
// SPAPD='(TRK,(100,10),RLSE)' VA PAC DATABASE BACKUP 2
//*****
// * INPUT FOR OPTIONAL REPORTS *
// * COL 2 : 'OR' *
// * COL 8 : ' ' PACBASE STATISTICS PRINTING *
// * : 'N' NO PACBASE STATISTICS PRINTING *
// * COL 9 : ' ' PACBASE LIMITATIONS PRINTING *
// * : 'N' NO PACBASE LIMITATIONS PRINTING *
//*****
// INPUT 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=3440)
// VERIFY EXEC PGM=IDCAMS
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
// SYSPRINT DD SYSOUT=&OUT
// PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
// PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
// SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
// PTUBAS EXEC PGM=PTUBAS
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
// PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
// PAC7DS DD SYSOUT=&OUT
// SYSOUT DD SYSOUT=&OUT
// SYSUDUMP DD SYSOUT=&OUT
// PTU500 EXEC PGM=PTU500,COND=(0,NE,PTUBAS)
//*****
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// *: DD DSN=&VSAMCAT,DISP=SHR
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
// PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR,
// AMP=('BUFNI=1,BUFND=3')
// PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR,
// AMP=('BUFND=3')
// PAC7MB DD DSN=&SAVEMB,DISP=(OLD,PASS)
// PAC7PC DD DSN=&INDUN..&ROOT.&FILE.PC(+1),
// DISP=(,CATLG,DELETE),

```

DATABASE MANAGEMENT UTILITIES

SAVE: DATABASE BACKUP

2

2

SAVE: EXECUTION JCL

4

```
//          UNIT=&UNITS ,
//          VOL=&VOLS ,
//          SPACE=&SPAPC ,
//          DCB=( &DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7PD   DD DSN=&INDUN. .&ROOT.&FILE.PD(+1) ,
//          DISP=( ,CATLG,DELETE) ,
//          UNIT=&UNITS2 ,
//          VOL=&VOLS2 ,
//          SPACE=&SPAPD ,
//          DCB=( &DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//PAC7DS   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELPD   EXEC PGM=IEFBR14,COND=(8,NE,PTU500)
//*****
//DDPC    DD DSN=&INDUN. .&ROOT.&FILE.PC(+1) ,DISP=(OLD,DELETE)
//DDPD    DD DSN=&INDUN. .&ROOT.&FILE.PD(+1) ,DISP=(OLD,DELETE)
```

DATABASE MANAGEMENT UTILITIES	PAGE	36
SASY: DATABASE SYSTEM BACKUP COMPLEMENT		2
SASY: INTRODUCTION		3
		1

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.

	PAGE	37
DATABASE MANAGEMENT UTILITIES		2
SASY: DATABASE SYSTEM BACKUP COMPLEMENT		3
SASY: INTRODUCTION		1

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.

DATABASE MANAGEMENT UTILITIES	
SASY: DATABASE SYSTEM BACKUP COMPLEMENT	
SASY: DESCRIPTION OF STEPS	

2
3
2

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=&INDUV..&ROOT.&FILE.AR
-Error message file
PAC7AE: DSN=&INDSV..&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=&INDUV..&ROOT.&FILE.AR

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

.Output Report:
-Review
PAC7GZ

DATABASE MANAGEMENT UTILITIES

2

SASY: DATABASE SYSTEM BACKUP COMPLEMENT

3

SASY: EXECUTION JCL

3

2.3.3. SASY: EXECUTION JCL

```

//*****
//* VisualAge Pacbase *
//* *
//* - COMPLEMENT OF 'SYSTEM' BACKUP OF THE DATABASE- *
//*****
//$RADP.SASY PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSN='$INDSN', INDEX OF NON VSAM SYSTEM FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT OUTPUT CLASS
//*****
//VERIFY EXEC PGM=IDCAMS
//*****
//*: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PTUBAS EXEC PGM=PTUBAS
//*****
//*: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7DS DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PTU502 EXEC PGM=PTU502,COND=(0,NE,PTUBAS)
//*****
//*: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7GZ DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	40
DATABASE MANAGEMENT UTILITIES		2
REST: DATABASE RESTORATION		4
REST: INTRODUCTION		1

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.

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
! ! ! 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.
+-----+

```

	PAGE	42
DATABASE MANAGEMENT UTILITIES		2
REST: DATABASE RESTORATION		4
REST: USER INPUT		2

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.

	PAGE	43
DATABASE MANAGEMENT UTILITIES	2	
REST: DATABASE RESTORATION	4	
REST: USER INPUT	2	

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.

	PAGE	44
DATABASE MANAGEMENT UTILITIES		2
REST: DATABASE RESTORATION		4
REST: USER INPUT		2

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.

REPORT RESULTS

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.

DATABASE MANAGEMENT UTILITIES	
REST: DATABASE RESTORATION	
REST: DESCRIPTION OF STEPS	

2
4
3

2.4.3. REST: DESCRIPTION OF STEPS

REST: DESCRIPTION OF STEPS

NOTE:

In the installation set, the &ARCHJRNL parameter is set to NULLFILE as a default. For a standard operation of the Database, this parameter must include the archived-transaction journal DSNNAME, in the following format:

```
ARCHJRNL='$indun..$root.$file'
```

so as to allow for the retrieval of archived transactions (REC parameter specified in the procedure's parameter card).

USER INPUT RECOGNITION: PTU004

```
.Input file:
  CARTE

.Output file:
  PAC7MB: DSN=&&MB

.Permanent input file:
-Error message file
  PAC7AE: DSN=&INDSV..&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=&INDSV..&ROOT.&ROOT.AE
-Journal file
  PAC7AJ: DSN=&INDUV..&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.
```

DATABASE MANAGEMENT UTILITIES

PAGE

46

REST: DATABASE RESTORATION

2

REST: DESCRIPTION OF STEPS

4

3

- 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=&INDUV..&ROOT.&FILE.AR
- Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
- Journal file
PAC7AJ: DSN=&INDUV..&ROOT.&FILE.AJ

DATABASE MANAGEMENT UTILITIES
REST: DATABASE RESTORATION
REST: DESCRIPTION OF STEPS

2
4
3

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=&INDSV..&ROOT.&ROOT.AE
-Sequential image of the database
 PAC7PC: DSN=&INDUN..&ROOT.&FILE.PC(0)
If backup option Dispatch:
-Sequential image of database 12
 PAC7PD: DSN=&INDUN..&ROOT.&FILE.PD(0)

.Permanent output files:
-Data file
 PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
-Index File
 PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
-Journal file
 PAC7AJ: DSN=&INDUV..&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 MANAGEMENT UTILITIES
REST: DATABASE RESTORATION
REST: DESCRIPTION OF STEPS

2
4
3

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=&INDUV..&ROOT.&FILE.AR

.Permanent input files:

-Journal to apply
PAC7JO: DSN=&ARCHJRNL
-Error message file
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

.Input work file:

PAC7PS: DSN=&&PAC7PS

DATABASE MANAGEMENT UTILITIES
REST: DATABASE RESTORATION
REST: DESCRIPTION OF STEPS

PAGE

49

2
4
3

.Output file:
-Update transactions
PAC70J: DSN=&&PAC70J 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 MANAGEMENT UTILITIES

2

REST: DATABASE RESTORATION

4

REST: DESCRIPTION OF STEPS

3

DATABASE UPDATE: PACA15

.Permanent update files:

-Data file

ARLSR: DSN=&INDUV..&ROOT.&FILE.AR
PAC7AR: SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')

-Index file

ANLSR: DSN=&INDUV..&ROOT.&FILE.AN
PAC7AN:
SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')

-Journal file

PAC7AJ: DSN=&INDUV..&ROOT.&FILE.AJ

.Permanent input files:

-Error message file

PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

-DSMS file of VA Pac elements

PAC7DC: DSN=&DSMS

(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

Note: If it is not possible to use the LSR batch system option in the PACA15 program, delete the LSR batch calls (DD SUBSYS), and replace ANLSR with PAC7AN, and ARLSR with PAC7AR.

DATABASE MANAGEMENT UTILITIES
 REST: DATABASE RESTORATION
 REST: EXECUTION JCL

2
 4
 4

2.4.4. REST: EXECUTION JCL

```

//*****
//* VisualAge Pacbase
//*
//* - RELOADING RESTORATION OF THE DATABASE -
//*****
//$RADP.REST PROC FILE=$FILE,          NUMBER OF PHYSICAL DATABASE
//      ROOT=$ROOT,                    ROOT OF THE VA PAC SYSTEM
//      INDSV='$INDSV',                 INDEX OF SYSTEM VSAM FILES
//      INDUV='$INDUV',                 INDEX OF USER VSAM FILES
//      INDUN='$INDUN',                 INDEX OF USER NON VSAM FILES
===SEQ FOR DSM
//      DSMS='$DSMS',                  DSNAME OF VA PAC-ELEMENTS DSMS FILE
===SEQ
//*:      VSAMCAT='$VCAT',              USER VSAM CATALOG
//*:      SYSTCAT='$SCAT',              VA PAC SYSTEM VSAM CATALOG
//      STEPLIB='$MODB',                LIBRARY OF LOAD-MODULES
//      LSR='BLSR',                     LSR BATCH SYSTEM NAME
//      OUT=$OUT,                       OUTPUT CLASS
//      ARCHJRNL=NULLFILE,             DSN OF JOURNAL TO APPLY
//      UWK=$UWK,                       WORK UNIT
//      SPAAJ='(TRK,(10,5))'           EXTRACTED JOURNAL SPACE
//*****
//* INPUT
//* COL 2      : 'Y'
//* COL 3-7    : NUMBER OF GAPS IN ABSOLUTE VALUE
//* COL 8-9    : NUMBER OF GAPS IN PERCENTAGE ( / BASE )
//* COL 10     : INITIAL LANGUAGE CODE (F=FRENCH, E=ENGLISH)
//* COL 11     : '1' INHIBITION OF TRANSACTION LOG
//* COL 12     : MACHINE DATE FORMAT ('N' FOR DD/MM/YY)
//*              ('I' FOR MM/DD/YY)
//* COL 13-15  : 'REC' FOR RECOVERY 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 35-46  : PF-KEYS SIGNIFICATIONS
//* COL 79     : BACKUP FILES DISPATCH
//*              : 'N' (DEFAULT VALUE) : NO DISPATCH (1 FILE)
//*              : 'D' : DISPATCH (2 FILES)
//*
//* IN THE ABSENCE OF INPUT, THE RELOAD DOES NOT MODIFY THE NUMBER
//* OF EXISTING GAPS, AND OTHER DATA IS UNCHANGED.
//*
//* IF THE JOURNAL FILE OF TRANSACTIONS ON DISK (AJ) IS NOT
//* REINITIALIZED, THE RESTORE CHAIN IS NOT EXECUTED.
//* IT IS THEREFORE NECESSARY TO EXECUTE THE ARCH PROCEDURE FIRST.
//*****
//INPUT EXEC PGM=PTU004
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&MB,DISP=(,PASS),
//      UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//EXISAJ EXEC PGM=IDCAMS,COND=(0,NE,INPUT)
//*****
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&ROOT.&FILE.SY(LI&ROOT.&FILE.AJ),DISP=SHR
//PTU380 EXEC PGM=PTU380,COND=((0,NE,INPUT),(0,NE,EXISAJ))
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR

```

DATABASE MANAGEMENT UTILITIES

REST: DATABASE RESTORATION

REST: EXECUTION JCL

2

4

4

```

//PAC7AJ DD DSN=&INDUV. .&ROOT.&FILE.AJ,DISP=SHR
//PAC7EU DD SYSOUT=&OUT
//DEFINE EXEC PGM=IDCAMS,COND=((0,NE,INPUT),(0,NE,PTU380))
//*****
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN. .&ROOT.&FILE.SY(DF&ROOT.&FILE.AN),DISP=SHR
// DD DSN=&INDUN. .&ROOT.&FILE.SY(DF&ROOT.&FILE.AR),DISP=SHR
// DD DSN=&INDUN. .&ROOT.&FILE.SY(DF&ROOT.&FILE.AJ),DISP=SHR
//PTU400 EXEC PGM=PTU400,COND=((0,NE,INPUT),(0,NE,PTU380))
//*****
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV. .&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ DD DSN=&INDUV. .&ROOT.&FILE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV. .&ROOT.&FILE.AN,DISP=SHR,
// AMP=('BUFNI=1,BUFND=4')
//PAC7AR DD DSN=&INDUV. .&ROOT.&FILE.AR,DISP=SHR,
// AMP=('BUFND=4')
//PAC7EU DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&MB,DISP=(OLD,DELETE)
//PAC7PC DD DSN=&INDUN. .&ROOT.&FILE.PC(0),DISP=OLD
//PAC7PD DD DSN=&INDUN. .&ROOT.&FILE.PD(0),DISP=SHR
//PAC7PS DD DSN=&&PAC7PS,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
// DCB=(RECFM=FB,LRECL=140,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT
//PTU420 EXEC PGM=PTU420,
// COND=((0,NE,INPUT),(0,NE,PTU380),(0,NE,PTU400))
//*****
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7JO DD DSN=&ARCHJRN,DISP=OLD,DCB=BLKSIZE=6179
//PAC7AE DD DSN=&INDSV. .&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV. .&ROOT.&FILE.AR,DISP=SHR
//PAC7OJ DD DSN=&&PAC7OJ,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAAJ,
// DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
//PAC7PS DD DSN=&&PAC7PS,DISP=(OLD,DELETE)
//PAC7EU DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACA15 EXEC PGM=PACA15,
// COND=((0,NE,INPUT),(0,NE,PTU380),(0,NE,PTU400),(0,NE,PTU420))
//*****
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7MV DD DSN=&&PAC7OJ,DISP=(OLD,DELETE)
//ANLSR DD DSN=&INDUV. .&ROOT.&FILE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV. .&ROOT.&FILE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7AJ DD DUMMY
//PAC7AE DD DSN=&INDSV. .&ROOT.&ROOT.AE,DISP=SHR
===SEQ FOR DSM
//PAC7DC DD DSN=&DSMS,DISP=SHR
===SEQ
//PAC7IE DD SYSOUT=&OUT
//PAC7IF DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	53
DATABASE MANAGEMENT UTILITIES		2
RESY: DATABASE SYSTEM RESTORATION COMPLEMENT		5
RESY: INTRODUCTION		1

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.

	PAGE	54
DATABASE MANAGEMENT UTILITIES		2
RESY: DATABASE SYSTEM RESTORATION COMPLEMENT		5
RESY: INTRODUCTION		1

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.

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)

DATABASE MANAGEMENT UTILITIES

2

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

5

RESY: USER INPUT - RESULTS

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 making it impossible to select archived transactions by date (PACX EXPJ procedure), 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 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.

	PAGE	57
DATABASE MANAGEMENT UTILITIES	2	
RESY: DATABASE SYSTEM RESTORATION COMPLEMENT	5	
RESY: USER INPUT - RESULTS	2	

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.

DATABASE MANAGEMENT UTILITIES	
RESY: DATABASE SYSTEM RESTORATION COMPLEMENT	
RESY: DESCRIPTION OF STEPS	

2
5
3

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=&INDSV..&ROOT.&ROOT.AE
-Journal file
PAC7AJ: DSN=&INDUV..&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 MANAGEMENT UTILITIES	
RESY: DATABASE SYSTEM RESTORATION COMPLEMENT	
RESY: DESCRIPTION OF STEPS	

2
5
3

DATABASE POSITIONING: PTU402

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

- .Permanent output file:
 - Data file
 - PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
- .Permanent input file:
 - Error message file
 - PAC7AE: DSN=&INDSV..&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=&INDUV..&ROOT.&FILE.AR
- .Permanent input files:
 - Journal to apply
 - PAC7JO: DSN=&ARCHJRNL
 - Error message file
 - PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
- .Input work file:
 - PAC7PS: DSN=&&PAC7PS

DATABASE MANAGEMENT UTILITIES

2

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

5

RESY: DESCRIPTION OF STEPS

3

.Output file:
 -Update transactions
 PAC70J: DSN=&&PAC70J 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
 ARLSR: DSN=&INDUV..&ROOT.&FILE.AR
 PAC7AR: SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
 -Index file
 ANLSR: DSN=&INDUV..&ROOT.&FILE.AN
 PAC7AN:
 SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
 -Journal file
 PAC7AJ: DSN=&INDUV..&ROOT.&FILE.AJ

.Permanent input files:
 -Error message file
 PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
 -DSMS file of VA Pac elements
 PAC7DC: DSN=&DSMS
 (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

Note: If it is not possible to use the LSR batch system option in the PACA15 program, delete the LSR batch calls (DD SUBSYS), and replace ANLSR with PAC7AN, and ARLSR with PAC7AR.

DATABASE MANAGEMENT UTILITIES

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

RESY: EXECUTION JCL

2

5

4

2.5.4. RESY: EXECUTION JCL

```

//*****
//* VisualAge Pacbase
//*
//* - 'SYSTEM' RELOADING RESTORATION COMPLEMENT -
//*****
//$RADP.RESY PROC FILE=$FILE,          NUMBER OF PHYSICAL DATABASE
//      ROOT=$ROOT,                    ROOT OF THE VA PAC SYSTEM
//      INDSV='$INDSV',                 INDEX OF SYSTEM VSAM FILES
//      INDUN='$INDUN',                 INDEX OF USER NON VSAM FILES
//      INDUV='$INDUV',                 INDEX OF USER VSAM FILES
===SEQ FOR DSM
//      DSMS='$DSMS',                  DSNAME OF VA PAC-ELEMENTS DSMS FILE
===SEQ
//*:      VSAMCAT='$VCAT',              USER VSAM CATALOG
//*:      SYSTCAT='$SCAT',              VA PAC SYSTEM VSAM CATALOG
//      STEPLIB='$MODB',                LIBRARY OF LOAD-MODULES
//      LSR='BLSR',                     LSR BATCH SYSTEM NAME
//      OUT=$OUT,                       OUTPUT CLASS
//      ARCHJRNL=NULLFILE,              DSN OF JOURNAL TO APPLY
//      UWK=$UWK,                       WORK UNIT
//      SPAAJ='(TRK,(10,5))'           EXTRACTED JOURNAL SPACE
//*****
//* INPUT
//* COL 2      : 'Y'
//* COL 10     : INITIAL LANGUAGE CODE (F=FRENCH, E=ENGLISH)
//* COL 11     : '1' INHIBITION OF TRANSACTION LOG
//* COL 12     : MACHINE DATE FORMAT ('N' FOR DD/MM/YY)
//*              ('I' FOR MM/DD/YY)
//* COL 13-15  : 'REC' FOR RECOVERY OF ARCHIVED TRANSACTIONS
//* COL 16-19  : 4 CHARACTERS TO BE DISPLAYED ON ALL PACBASE 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 35-46  : PF-KEYS SIGNIFICATIONS
//* COL 79     : BACKUP FILES DISPATCH
//*              : 'N' (DEFAULT VALUE) : NO DISPATCH (1 FILE)
//*              : 'D' : DISPATCH (2 FILES)
//*
//* IF THE JOURNAL FILE OF TRANSACTIONS ON DISK (AJ) IS NOT
//* REINITIALIZED, THE RESTORE CHAIN IS NOT EXECUTED.
//* IT IS THEREFORE NECESSARY TO EXECUTE THE ARCH PROCEDURE FIRST.
//*****
//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=3440)
//EXISAJ EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN.&ROOT.&FILE.SY(LI&ROOT.&FILE.AJ),DISP=SHR
//PTU380 EXEC PGM=PTU380,COND=(0,NE,EXISAJ)
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV.&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ DD DSN=&INDUV.&ROOT.&FILE.AJ,DISP=SHR
//PAC7EU DD SYSOUT=&OUT
//PTU402 EXEC PGM=PTU402,COND=(0,NE,PTU380)
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV.&ROOT.&ROOT.AE,DISP=SHR

```

DATABASE MANAGEMENT UTILITIES

2

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

5

RESY: EXECUTION JCL

4

```

//PAC7AR DD DSN=&INDUV. .&ROOT.&FILE.AR,DISP=SHR
//PAC7GZ DD SYSOUT=&OUT
//PAC7MB DD DSN=&&RESYMB,DISP=(OLD,DELETE)
//PAC7PS DD DSN=&&PAC7PS,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//      DCB=(RECFM=FB,LRECL=140,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT
//PTU420 EXEC PGM=PTU420,COND=((0,NE,PTU380),(0,NE,PTU402))
//*****
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7JO DD DSN=&ARCHJRN,DISP=OLD,DCB=BLKSIZE=6179
//PAC7AE DD DSN=&INDSV. .&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV. .&ROOT.&FILE.AR,DISP=SHR
//PAC7OJ DD DSN=&&PAC7OJ,DISP=(,PASS),
//      UNIT=&UWK,SPACE=&SPAAJ,
//      DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179)
//PAC7PS DD DSN=&&PAC7PS,DISP=(OLD,DELETE)
//PAC7EU DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACA15 EXEC PGM=PACA15,
//      COND=((0,NE,PTU380),(0,NE,PTU402),(0,NE,PTU420))
//*****
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7MV DD DSN=&&PAC7OJ,DISP=(OLD,DELETE)
//ANLSR DD DSN=&INDUV. .&ROOT.&FILE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV. .&ROOT.&FILE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7AJ DD DUMMY
//PAC7AE DD DSN=&INDSV. .&ROOT.&ROOT.AE,DISP=SHR
===SEQ FOR DSM
//PAC7DC DD DSN=&DSMS,DISP=SHR
===SEQ
//PAC7IE DD SYSOUT=&OUT
//PAC7IF DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

DATABASE MANAGEMENT UTILITIES	PAGE	63
ARCH: JOURNAL ARCHIVAL		2
ARCH: INTRODUCTION		6
		1

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 transactions 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.)

	PAGE	65
DATABASE MANAGEMENT UTILITIES	2	
ARCH: JOURNAL ARCHIVAL	6	
ARCH: INPUT - RECOMMENDATIONS - RESULTS	2	

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 'T' must be entered in column 15. As a result, the ARCH procedure formats a new Journal Back-up file.

	PAGE	66
DATABASE MANAGEMENT UTILITIES		2
ARCH: JOURNAL ARCHIVAL		6
ARCH: INPUT - RECOMMENDATIONS - RESULTS		2

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:

- . 'T' 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 subsequently 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.

DATABASE MANAGEMENT UTILITIES	
ARCH: JOURNAL ARCHIVAL	
ARCH: DESCRIPTION OF STEPS	

PAGE	67
	2
	6
	3

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=&INDSV..&ROOT.&ROOT.AE
-Previously archived transactions
PAC7JP: DSN=&INDUN..&ROOT.&FILE.PJ(0)
-Journal file to reinitialize
PAC7AJ: DSN=&INDUV..&ROOT.&FILE.AJ

.Input work file:

-User transaction
PAC7MB: DSN=&&ARCHMB

.Permanent input-Output file:

-Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR

.Output files:

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

DATABASE MANAGEMENT UTILITIES	
ARCH: JOURNAL ARCHIVAL	
ARCH: DESCRIPTION OF STEPS	

2
6
3

.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=&INDUV..&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=&INDUV..&ROOT.&FILE.AR

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

DATABASE MANAGEMENT UTILITIES

PAGE

69

ARCH: JOURNAL ARCHIVAL

2

ARCH: DESCRIPTION OF STEPS

6

3

.Output file:
-Journal file to re-initialize
PAC7AJ: DSN=&INDUV..&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=&INDUN..&ROOT.&FILE.PJ(+1)

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

DATABASE MANAGEMENT UTILITIES

ARCH: JOURNAL ARCHIVAL

ARCH: EXECUTION JCL

2

6

4

2.6.4. ARCH: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - ARCHIVAL OF THE JOURNAL - *
//*****
// $RADP.ARCH PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSN='$INDSN', INDEX OF SYSTEM NON VSAM FILES
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDUV='$INDUV', INDEX OF USER VSAM FILES
// INDUN='$INDUN', INDEX OF USER NON VSAM FILES
// *: VSAMCAT='$VCAT', USER VSAM CATALOG
// *: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// DSCB='$DSCB', DSCB MODEL FILE
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', VOLUME OF ARCHIVED JOURNAL
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// UWK=$UWK, WORK UNIT
// SPAPJ='(TRK,(100,10),RLSE)' SPACE OF JOURNAL
//*****
// * INPUT : COMMAND FOR DEACTIVATION OF ARCHIVED TRANSACTIONS *
// * COL 2 : 'S' *
// * COL 3 TO 6 : SESSION NUMBER *
// * COL 7 TO 14 : DATE (CCYYMMDD) *
// * COL 15 : ' ' PRESENCE OF ARCHIVED TRANSACTION FILE *
// * : 'I' ABSENCE OF ARCHIVED TRANSACTION FILE *
// * COL 16 : ' ' PRESENCE OF DATA FILE (AR) *
// * : 'D' ABSENCE OF DATA FILE (AR) *
// * COL 17 : ' ' ARCHIVAL AND REINITIALIZATION *
// * : 'J' REINITIALIZATION WITHOUT ARCHIVAL *
// * *
// * IN THE ABSENCE OF INPUT (OR ERROR ON A COMMAND PARAMETER) *
// * NO DEACTIVATION WILL TAKE PLACE, HOWEVER ARCHIVAL AND *
// * REINITIALIZATION WILL BE EXECUTED NORMALLY. *
// * *
// * TRANSACTIONS WHOSE SESSION (DATE) IS PRIOR OR EQUAL TO *
// * THE SESSION (DATE) INDICATED ARE NOT KEPT. THEY ARE *
// * RECOVERED IN THE FILE OF DEACTIVATED TRANSACTION. *
// * *
//*****
// 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=3440)
// VERIFY EXEC PGM=IDCAMS
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
// SYSPRINT DD SYSOUT=&OUT
// PAC7AJ DD DSN=&INDUV.&ROOT.&FILE.AJ,DISP=SHR
// PAC7AR DD DSN=&INDUV.&ROOT.&FILE.AR,DISP=SHR
// SYSIN DD DSN=&INDSN.&ROOT.&ROOT.SY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN.&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
// PTU300 EXEC PGM=PTU300
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// PAC7MB DD DSN=&&ARCHMB,DISP=(OLD,PASS)
// PAC7AE DD DSN=&INDSV.&ROOT.&ROOT.AE,DISP=SHR
// PAC7AR DD DSN=&INDUV.&ROOT.&FILE.AR,DISP=SHR
// PAC7AJ DD DSN=&INDUV.&ROOT.&FILE.AJ,DISP=SHR
// PAC7JP DD DSN=&INDUN.&ROOT.&FILE.PJ(0),DISP=OLD
// PAC7PJ DD DSN=&INDUN.&ROOT.&FILE.PJ(+1),
// DISP=(,CATLG,DELETE),

```

DATABASE MANAGEMENT UTILITIES

ARCH: JOURNAL ARCHIVAL

ARCH: EXECUTION JCL

2

6

4

```

//          UNIT=&UNITS ,
//          VOL=&VOLS ,
//          SPACE=&SPAPJ ,
//          DCB=( &DSCB,RECFM=FB,LRECL=167,BLKSIZE=6179)
//PAC7PQ   DD DUMMY,DCB=BLKSIZE=167
//PAC7EU   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELDEF  EXEC PGM=IDCAMS,COND=(0,NE,PTU300)
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDUN..&ROOT.&FILE.SY(DF&ROOT.&FILE.AJ),DISP=SHR
//PTU320   EXEC PGM=PTU320,COND=(0,NE,PTU300)
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&ARCHMB,DISP=(OLD,DELETE)
//PAC7AE   DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&ROOT.&FILE.AJ,DISP=SHR
//PAC7EU   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELPJ    EXEC PGM=IEFBR14,COND=(8,NE,PTU300)
//*****
//DDPJ     DD DSN=&INDUN..&ROOT.&FILE.PJ(+1),DISP=(OLD,DELETE)

```

	PAGE	72
DATABASE MANAGEMENT UTILITIES		2
REOR: DATABASE REORGANIZATION		7
REOR: INTRODUCTION		1

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.

	PAGE	73
DATABASE MANAGEMENT UTILITIES		2
REOR: DATABASE REORGANIZATION		7
REOR: INTRODUCTION		1

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

DATABASE MANAGEMENT UTILITIES
 REOR: DATABASE REORGANIZATION
 REOR: INPUT - RECOMMENDATIONS

2
 7
 2

```

-----
!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 !
-----
  
```

Maximum number of occurrences by type: 2,500.

The 'List of ignored 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.

```

+-----+
!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)

	PAGE	76
DATABASE MANAGEMENT UTILITIES	2	
REOR: DATABASE REORGANIZATION	7	
REOR: INPUT - RECOMMENDATIONS	2	

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

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=&INDSV..&ROOT.&ROOT.AE
-Sequential image of the database
PAC7PC: DSN=&INDUN..&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: SRTRE01 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

DATABASE MANAGEMENT UTILITIES
REOR: DATABASE REORGANIZATION
REOR: DESCRIPTION OF STEPS

PAGE

78

2
7
3

PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

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

.Output report:
-Entity-purge transactions

DATABASE MANAGEMENT UTILITIES	
REOR: DATABASE REORGANIZATION	
REOR: DESCRIPTION OF STEPS	

2
7
3

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=&INDSV..&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.

DATABASE MANAGEMENT UTILITIES

2

REOR: DATABASE REORGANIZATION

7

REOR: DESCRIPTION OF STEPS

3

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=&INDSV..&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

DATABASE MANAGEMENT UTILITIES
REOR: DATABASE REORGANIZATION
REOR: DESCRIPTION OF STEPS

PAGE

81

2
7
3

.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=&INDSV..&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=&INDUN..&ROOT.&FILE.PC(+1)
If Dispatch option of backup:
-Sequential image of the database Í2
PAC7PD: DSN=&INDUN..&ROOT.&FILE.PD(+1)

.Output report:
-Logical database building
PAC7IE

DATABASE MANAGEMENT UTILITIES
 REOR: DATABASE REORGANIZATION
 REOR: EXECUTION JCL

2
 7
 4

2.7.4. REOR: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - REORGANIZATION OF THE DATABASE - *
//*****
// $RADP.REOR PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSN='$INDSN', INDEX OF NON VSAM SYSTEM FILES
// INDUN='$INDUN', INDEX OF NON VSAM USER FILES
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// *: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// SORTLIB='$BIBT', SORT LIBRARY
// DSCB='$DSCB', DSCB MODEL FILE
// OUT=$OUT, OUTPUT CLASS
// VOLS='$SER=$VOLUN', DATABASE BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// VOLS2='$SER=$VOLUN', DATABASE 2ND BACKUP VOLUME
// UNITS2=$UNITUN, 2ND BACKUP UNIT (DISK OR CARTRIDGE)
// UWK=$UWK, WORK FILES UNIT
// SPAPC='(TRK,(100,10),RLSE)', VA PAC DATABASE BACKUP
// SPAPD='(TRK,(100,10),RLSE)', VA PAC DATABASE BACKUP 2
// CYL='(10,1)' TEMPORARY SPACE
//*****
// 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=3440)
// PTU2CL EXEC PGM=PTU2CL
//*****
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// *: STEPCAT DD DSN=&SYSTCAT,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)
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
// SYSUDUMP DD SYSOUT=&OUT
// PTU200 EXEC PGM=PTU200,COND=(0,NE,PTU2CL)
//*****
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// PAC7EE DD SYSOUT=&OUT
// SYSUDUMP DD SYSOUT=&OUT
// PAC7PC DD DSN=&INDUN..&ROOT.&FILE.PC(0),DISP=SHR
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
// PAC7PR DD DSN=&&QS,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=165,BLKSIZE=27225)
// SORTQS EXEC PGM=SORT,COND=((0,NE,PTU2CL),(0,NE,PTU200))
//*****
// SORTLIB DD DSN=&SORTLIB,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// SYSPRINT DD SYSOUT=&OUT
// SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
// SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
// SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
// SORTOUT DD DSN=&&SQ,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=165,BLKSIZE=27225)
// SORTIN DD DSN=&&QS,DISP=(OLD,DELETE)
// SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(SRTREO1),DISP=SHR
// PTU208 EXEC PGM=PTU208,
// COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS))
//*****

```

DATABASE MANAGEMENT UTILITIES

2

REOR: DATABASE REORGANIZATION

7

REOR: EXECUTION JCL

4

```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV. &ROOT.&ROOT.AE,DISP=SHR
//PAC7PU DD DSN=&&PU,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=13,BLKSIZE=6227)
//PAC7EE DD SYSOUT=&OUT
//PAC7MB DD DSN=&&REORBM,DISP=(OLD,PASS)
//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)
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PTU210 EXEC PGM=PTU210,
//          COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS))
//          *****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//PAC7PR DD DSN=&&SQ,DISP=(OLD,DELETE)
//PAC7PU DD DSN=&&PU,DISP=(OLD,DELETE)
//PAC7AE DD DSN=&INDSV. &ROOT.&ROOT.AE,DISP=SHR
//PAC7UM DD DSN=&&UM,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=165,BLKSIZE=6105)
//PAC7QS DD DSN=&&QS,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=165,BLKSIZE=27225)
//PAC7EB DD SYSOUT=&OUT
//PAC7EE DD SYSOUT=&OUT
//PAC7EK DD SYSOUT=&OUT
//PAC7MB DD DSN=&&REORBM,DISP=(OLD,PASS)
//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)
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PTU220 EXEC PGM=PTU220,REGION=4096K,
//          COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),(0,NE,PTU210))
//          *****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7EE DD SYSOUT=&OUT
//PAC7MR DD UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=140,BLKSIZE=6160)
//PAC7AE DD DSN=&INDSV. &ROOT.&ROOT.AE,DISP=SHR
//PAC7UM DD DSN=&&UM,DISP=(OLD,DELETE)
//PAC7UR DD DSN=&&QS,DISP=(OLD,DELETE)
//PAC7AN DD DSN=&&AN,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=55,BLKSIZE=27225)
//PAC7PA DD DSN=&&PA,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=149,BLKSIZE=27267)
//PAC7PB DD DSN=&&PB,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=149,BLKSIZE=27267)
//PAC7PC DD DSN=&&PC,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(TRK,1,RLSE),
//          DCB=(RECFM=FB,LRECL=149,BLKSIZE=149)
//SYSUDUMP DD SYSOUT=&OUT
//SORTAN EXEC PGM=SORT,
//          COND=((0,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),(0,NE,PTU210))
//          *****
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN DD DSN=&&AN,DISP=(OLD,DELETE)
//SORTOUT DD DSN=&&NA,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),

```

DATABASE MANAGEMENT UTILITIES

2

REOR: DATABASE REORGANIZATION

7

REOR: EXECUTION JCL

4

```

//          DCB=(RECFM=FB,LRECL=55,BLKSIZE=27225)
//SYSIN    DD DSN=&INDSN. &ROOT.&ROOT.SY(SRTREO2),DISP=SHR
//PTU240 EXEC PGM=PTU240,COND=((0,NE,PTU2CL),(0,NE,PTU200),
//          (0,NE, SORTQS),(0,NE,PTU210),(0,NE, SORTAN))
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7IE DD SYSOUT=&OUT
//PAC7MB DD DSN=&&REORBM,DISP=(OLD,DELETE)
//PAC7AE DD DSN=&INDSV. &ROOT.&ROOT.AE,DISP=SHR
//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=&INDUN. &ROOT.&FILE.PC(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,
//          VOL=&VOLS,
//          SPACE=&SPAPC,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7PD DD DSN=&INDUN. &ROOT.&FILE.PD(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS2,
//          VOL=&VOLS2,
//          SPACE=&SPAPD,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//SYSUDUMP DD SYSOUT=&OUT

```

DATABASE MANAGEMENT UTILITIES	PAGE	85
SVAG: GENERATION-PRINT REQUEST BACKUP		2
SVAG: INTRODUCTION		8
		1

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.

DATABASE MANAGEMENT UTILITIES	
SVAG: GENERATION-PRINT REQUEST BACKUP	
SVAG: DESCRIPTION OF STEPS	

2
8
2

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=&INDUV..&ROOT.&FILE.AG
-Error messages
 PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
-User input
 PAC7MB: DSN=&&SVAGMB

.Output file:
-Sequential image of requests
 PAC7PG: DSN=&INDUN..&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.

DATABASE MANAGEMENT UTILITIES

SVAG: GENERATION-PRINT REQUEST BACKUP

SVAG: EXECUTION JCL

2

8

3

2.8.3. SVAG: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - BACKUP OF 'AG' FILE - *
//*****
//$RADP.SVAG PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUN='$INDUN', INDEX OF NON-VSAM USER FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// UNITS=$UNITUN, UNIT OF BACKUP
// VOLS='SER=$VOLUN', VOLUME OF BACKUP
// UWK=$UWK, WORK UNIT
// DSCB='$DSCB', DSCB MODEL FILE
// SPAPG='(TRK,(20,5),RLSE)', SPACE OF BACKUP
// OUT=$OUT OUTPUT CLASS
//*****
// * NO INPUT *
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&SVAGMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7AG DD DSN=&INDUV..&ROOT.&FILE.AG,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAG),DISP=SHR
//PTU550 EXEC PGM=PTU550
//*****
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7MB DD DSN=&&SVAGMB,DISP=(OLD,DELETE)
//PAC7AG DD DSN=&INDUV..&ROOT.&FILE.AG,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7PG DD DSN=&INDUN..&ROOT.&FILE.PG(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPG,
// DCB=(&DSCB,RECFM=FB,LRECL=150,BLKSIZE=6150)
//PAC7EE DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//DELPG EXEC PGM=IEFBR14,COND=(8,NE,PTU550)
//*****
//DDPG DD DSN=&INDUN..&ROOT.&FILE.PG(+1),DISP=(OLD,DELETE)

```

DATABASE MANAGEMENT UTILITIES	PAGE	88
REAG: GENERATION-PRINT REQUEST RESTORATION		2
REAG: INTRODUCTION		9
		1

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 !	!uuuuuuu!	! User to be purged	('AU')!

Maximum number of sessions.....: 500

Maximum number of libraries.....: 100

Maximum number of users.....: 100

Default option: restoration.

DATABASE MANAGEMENT UTILITIES	
REAG: GENERATION-PRINT REQUEST RESTORATION	
REAG: DESCRIPTION OF STEPS	

PAGE	90
	2
	9
	3

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=&INDSV..&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=&INDUV..&ROOT.&FILE.AG
```

INITIALIZATION-REORGANIZATION OF REQUEST FILE (AG): PTU560

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

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

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

	PAGE	91
DATABASE MANAGEMENT UTILITIES		2
REAG: GENERATION-PRINT REQUEST RESTORATION		9
REAG: DESCRIPTION OF STEPS		3

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

DATABASE MANAGEMENT UTILITIES

REAG: GENERATION-PRINT REQUEST RESTORATION

REAG: EXECUTION JCL

2

9

4

2.9.4. REAG: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - INITIALIZATION-RESTORATION OF 'AG' FILE - *
//*****
// $RADP.REAG PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDUN='$INDUN', INDEX OF NON-VSAM USER FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// *: VSAMCAT='$VCAT', USER VSAM CATALOG
// *: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// UWK=$UWK, UNIT OF WORK FILE
// OUT=$OUT OUTPUT CLASS
//*****
// * REST. OR INIT. REQUEST ( DEFAULT OPTION = REST.) *
// * COL 2-3 : 'AG' *
// * COL 4 : ' ' FOR RESTORATION *
// * : 'I' FOR INITIALIZATION *
// * DELETION REQUESTS (OPTIONAL LINES) *
// * COL 2-6 : 'ABXXX' DELETION OF XXX LIBRARY'S COMMANDS *
// * COL 2-7 : 'ASXXX' DELETION OF XXXX SESSION'S COMMANDS *
// * COL 2-9 : 'AUXXXXXX' DELETION OF XXXXXX USER'S COMMANDS *
//*****
// INPUT EXEC PGM=PTU004
//*****
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// CARTE DD DDNAME=SYSIN
// PAC7AE DD DSN=&INDSV. &ROOT. &ROOT. AE,DISP=SHR
// PAC7DD DD SYSOUT=&OUT
// PAC7MB DD DSN=&&MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
// DEFINE EXEC PGM=IDCAMS,COND=(0,NE,INPUT)
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// SYSPRINT DD SYSOUT=&OUT
// SYSIN DD DSN=&INDUN. &ROOT. &FILE. SY(DF&ROOT. &FILE. AG),DISP=SHR
// PTU560 EXEC PGM=PTU560,COND=(0,NE,INPUT)
//*****
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// *: DD DSN=&VSAMCAT,DISP=SHR
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// PAC7AG DD DSN=&INDUV. &ROOT. &FILE. AG,DISP=SHR
// PAC7AE DD DSN=&INDSV. &ROOT. &ROOT. AE,DISP=SHR
// PAC7PG DD DSN=&INDUN. &ROOT. &FILE. PG(0),DISP=OLD
// PAC7MB DD DSN=&&MB,DISP=(OLD,DELETE)
// PAC7EE DD SYSOUT=&OUT
// PAC7DD DD SYSOUT=&OUT
// PAC7EK DD SYSOUT=&OUT
// SYSUDUMP DD SYSOUT=&OUT

```

DATABASE MANAGEMENT UTILITIES	PAGE	93
<i>PARM: UPDATE OF USER PARAMETERS</i>		2
<i>PARM: INTRODUCTION</i>		10
		1

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.

DATABASE MANAGEMENT UTILITIES
PARM: UPDATE OF USER PARAMETERS
PARM: INTRODUCTION

PAGE

94

2
10
1

	PAGE	95
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: INTRODUCTION		1

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).

DATABASE MANAGEMENT UTILITIES
PARM: UPDATE OF USER PARAMETERS
PARM: INPUT - RECOMMENDATIONS

2
10
2

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.

	PAGE	98
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: INPUT - RECOMMENDATIONS		2

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 ! 'OUI' ! Activation of the DSMS database control !
! ! ! ! ! (except for IBM MVS) !
! ! ! ! ' ' ! No DSMS control !
!-----!-----!-----!-----!
  
```

DATABASE MANAGEMENT UTILITIES

PARM: UPDATE OF USER PARAMETERS

2

10

PARM: INPUT - RECOMMENDATIONS

2

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 !
!-----!
  
```

	PAGE	102
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: INPUT - RECOMMENDATIONS		2

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.

DATABASE MANAGEMENT UTILITIES	PAGE	103
PARM: UPDATE OF USER PARAMETERS		2
PARM: USER-CODE DEFINITION		10
		3

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

	PAGE	104
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: USER-CODE DEFINITION		3

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

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 'API' 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 'API'.

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.

	PAGE	105
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: USER-CODE DEFINITION		3

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.

DATABASE MANAGEMENT UTILITIES
 PARM: UPDATE OF USER PARAMETERS
 PARM: USER-CODE GLOBAL AUTHORIZATIONS

PAGE 106
 2
 10
 4

2.10.4. PARM: USER-CODE GLOBAL AUTHORIZATIONS

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	8		<p>USER CODE (REQ. IN CREATION)</p> <p>Each user must be given a personal user code and associated password.</p> <p>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).</p> <p>The user code is stored for each transaction in the Journal.</p> <p>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.</p>
2	3	NUMER. 000	<p>LINE NUMBER (REQ. IN CREATION)</p> <p>General definition line of a user (code, password and global authorization). Used as the key.</p>
3	8		<p>USER PASSWORD</p> <p>The password is associated with a user code. Using blanks between two characters is forbidden.</p> <p>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.</p>
4	1	Blank 0 1 2 3	<p>GENERAL AUTHORIZATION LEVEL</p> <p>This authorization grants access to the Database.</p> <p>Blank No global access authorization.</p> <p>0 No global access authorization.</p> <p>1 Read-only access authorized for both current and all frozen sessions.</p> <p>2 Read-write access authorized for the current session and read-only access for all frozen sessions.</p> <p>3 Read-write access is authorized for both current and test sessions.</p> <p>NOTE: This authorization is limited by the provisions of the PROTECTION OF EXTRACTED ENTITIES and MODIFICATION OF EXTRACTED LINES fields on the</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		4	<p>Library Definition screen of the libraries concerned.</p> <p>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.</p>
5	1	NUMER. Blank 0 1 2 or 3 4	<p>USER-PARAMETER UPDATE AUTHORIZATION</p> <p>This level concerns authorizations for the user-parameter management access.</p> <p>Access prohibited.</p> <p>Access prohibited.</p> <p>Read-only access.</p> <p>Read-write access.</p> <p>Administrator's authorization.</p>
6	1	Blank 0 2 3 4	<p>GENERAL AUTHORIZATION ON PROCEDURES</p> <p>No authorization on the batch procedures.</p> <p>No authorization on the batch procedures (default option in creation)</p> <p>AUTHORIZATION ON STANDARD EXTRACTIONS</p> <p>Level allowing access to standard extractors.</p> <p>AUTHORIZATION ON SPECIAL EXTRACTIONS</p> <p>"Project Manager" level: Level granting access to special procedures.</p> <p>MAXIMUM AUTHORIZATION</p> <p>"VisualAge Pacbase Manager" level: Access to the database management, generation-print and PEI file management procedures.</p> <p>NOTE: This level can be granted for a global authorization only.</p>
7	30		<p>USER NAME</p> <p>Name may be entered in lower-case print.</p>

DATABASE MANAGEMENT UTILITIES
PARM: UPDATE OF USER PARAMETERS
PARM: USER-CODE GLOBAL AUTHORIZATIONS

PAGE 108
2
10
4

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
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		<p>USER CODE</p> <p>Each user must be given a personal user code and associated password.</p> <p>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).</p> <p>The user code is stored for each transaction in the Journal.</p> <p>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.</p>
2	3	1 to 999	<p>LINE NUMBER</p> <p>It is advisable to leave gaps in the line numbering sequence in order to facilitate future insertions.</p> <p>SPECIFIC AUTHORIZATION: - on libraries, - on the PEI function.</p>
3	4		<p>DATABASE CODE</p> <p>FOR MULTI-DATABASE SITES ONLY.</p> <p>Logical name of the database. This code is displayed in the identifier which appears in the top right corner of all screens.</p> <p>It is used to establish the relation between a VA-Pacbase database and a DSMS database.</p> <p>No validity check is performed here.</p>
			<p>LIBRARY ACCESS TABLE NOMBRE DE REPETITIONS : 15</p> <p>Two access types may be entered:</p> <ul style="list-style-type: none"> - Access to a Database library, - Access to the Production Environment Interface (PEI function).
	3	BBB	<p>LIBRARY CODE</p> <p>Code identifying the selected library.</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		***	Read-only access authorization on the whole database ('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.

DATABASE MANAGEMENT UTILITIES
 PARM: UPDATE OF USER PARAMETERS
 PARM: USER-CODE SPECIFIC AUTHORIZATIONS

PAGE 111
 2
 10
 5

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
7	1		BATCH PROCEDURE AUTHORIZATION LEVEL
		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.

DATABASE MANAGEMENT UTILITIES	PAGE	112
PARM: UPDATE OF USER PARAMETERS		2
PARM: TEXT TYPES		10
		6

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	F E	LANGUAGE INDICATOR French. English.
2	2	T	<p>TYPE OF TEXT (REQ. IN CREATION)</p> <p>The TYPE OF TEXT field is used for documentation purposes only, and allows the user to:</p> <ul style="list-style-type: none"> .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. <p>The coding of types and labels depends on an external parameter handled by the Database Administrator.</p> <p>Default value.</p>
3	15		<p>NAME OF TEXT TYPE (REQ. IN CREATION)</p> <p>Specify the label to appear with the corresponding Type of Text.</p> <p>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.</p> <p>Enter the name to appear with the corresponding Type of Text.</p> <p>This name will appear on the Text Definition screen when the corresponding Type of Text is used.</p>

	PAGE	114
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: MODIFICATION OF STANDARD ERROR MESSAGES		7

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 field identifies the type of error:
		2	Invalid absence of element
		3	Invalid presence of element
		4A	Element value is not alphabetic
		4Z	Element value is not numeric
		5	Element value content is not valid
		8F	Invalid creation - entity already exists
		9F	Invalid deletion/modification - entity does not exist
		9G	End of list
			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'
			CLIENT/SERVER FACILITY MESSAGES
		DUPL	To modify 'INVALID CREATION RECORD'
		NFND	To modify 'INVALID DELETE/MODIFY RECORD'
		END	To modify 'END OF LIST'

DATABASE MANAGEMENT UTILITIES
 PARM: UPDATE OF USER PARAMETERS
 PARM: MODIFICATION OF STANDARD ERROR MESSAGES

PAGE 116
 2
 10
 7

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		ABSC	To modify 'ABSENCE OF RECORD'
3	30	Blank	<p>FIRST PART OF ERROR MESSAGE (REQ. IN CREATION)</p> <p>Enter the message to appear before the erroneous data element name for the corresponding Error Code.</p> <p>Note: This message will be stored in the User Parameter file (AP).</p> <p>Default value with error type code "2": INVALID ABSENCE FOR THE FIELD</p> <p>Default value with error type code "3": INVALID PRESENCE FOR THE FIELD</p> <p>Default value with error type code "4A":</p> <p>This is the constant that will appear in the first part of the standard error message generated by the system. It is stored in the User Parameter file (AP).</p> <p>Default value with error type code "5": INVALID VALUE FOR THE FIELD</p> <p>Default value with error type code "8F": INVALID CREATION RECORD</p> <p>Default value with error type code "9F": INVALID DELETE/MODIFY RECORD</p> <p>Default value with error type code "9G": END OF LIST</p>

DATABASE MANAGEMENT UTILITIES	PAGE	117
PARM: UPDATE OF USER PARAMETERS		2
PARM: GENERATED-STREAM CONTROL CARDS		10
		8

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.

	PAGE	118
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: GENERATED-STREAM CONTROL CARDS		8

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.

	PAGE	119
DATABASE MANAGEMENT UTILITIES		2
PARM: UPDATE OF USER PARAMETERS		10
PARM: GENERATED-STREAM CONTROL CARDS		8

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.

DATABASE MANAGEMENT UTILITIES
PARM: UPDATE OF USER PARAMETERS
PARM: GENERATED-STREAM CONTROL CARDS

PAGE

120

2

10

8

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	A D F Z	TYPE OF OPTION (REQ. IN CREATION) Beginning of generated program job stream. Before the generated program. Following the generated program. 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	BLANK 0 - 99 NUMERICAL	LINE NUMBER (REQ. IN CREATION) Option title line: Title in the "Optional Card Image" field. Lower-case keying accepted. 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:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		A	Library code ('*' entity, 1 to 3 characters).
		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) jj/mm/aa (8) = Generation date or mm/jj/aa, 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 C/S dialog)
		R	Clear name of generated program, screen, or block (from definition screen).
		S	Code of generated program, screen or block.
		U	User code.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		V 1 to 9	<p>Job stream number (two-digit value), automatically assigned according to the order of execution.</p> <p>Numerical values of input parameters will be decoded according to the values on the GENERATION AND PRINT COMMANDS (GP) screen.</p> <p>NOTE: This field accepts lowercase characters.</p>
5	1		<p>INPUT PARAMETER NO.1</p> <p>Can take any one of the values as defined above.</p> <p>Can take on any of the values defined above as well as numerical values.</p>
6	1		<p>INPUT PARAMETER NO.2</p> <p>Can take any one of the values defined above.</p> <p>Can take on any of the values defined above as well as numerical values.</p>
7	1		<p>INPUT PARAMETER NO.3</p> <p>Can take any one of the values defined above.</p> <p>Can take on any of the values defined above as well as numerical values.</p>
8	1		<p>INPUT PARAMETER NO.4</p> <p>Can take any one of the values defined above.</p> <p>Can take on any of the values defined above as well as numerical values.</p>
9	1		<p>INPUT PARAMETER NO.5</p> <p>Can take any one of the values defined above.</p> <p>Can take on any of the values defined above as well as numerical values.</p>
10	1		<p>INSERTION REFERENCE CHARACTER</p> <p>This is a given character that will be replaced, in the generated control card, by the values of the input parameter codes.</p> <p>The first occurrence of this character is replaced by the field selected by the first non-blank input parameter.</p> <p>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.</p> <p>This is a given character that will be replaced, in</p>

DATABASE MANAGEMENT UTILITIES
 PARM: UPDATE OF USER PARAMETERS
 PARM: GENERATED-STREAM CONTROL CARDS

PAGE

124

2

10

8

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			<p>the generated control card, by the values of the input parameter codes.</p> <p>The first occurrence of this character is replaced by the field selected by the first non-blank input parameter.</p> <p>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).</p> <p>The second occurrence of this character is replaced by the field selected by the second non-blank input parameter.</p> <p>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.</p> <p>Insertion Reference Characters which have not been replaced, as well as those which correspond to an erroneous input parameter, will remain unchanged.</p>

DATABASE MANAGEMENT UTILITIES	
PARM: UPDATE OF USER PARAMETERS	
PARM: DESCRIPTION OF STEPS	

PAGE	125
	2
	10
	9

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=&INDSV..&ROOT.&ROOT.AE
-User parameters
 PAC7AP: DSN=&INDSV..&ROOT.&ROOT.AP

.Permanent input files:

-User parameter backup
 PAC7EC: DSN=&INDSN..&ROOT.&ROOT.PE(0)

.Transaction file:

-Update transactions
 PAC7MC: DSN=&&PARMMB

.Output file

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

.Output reports

-Printing of the update file and review
 PAC7IJ
-Check on procedure access authorization
 PAC7DD

DATABASE MANAGEMENT UTILITIES
PARM: UPDATE OF USER PARAMETERS
PARM: DESCRIPTION OF STEPS

2
10
9

.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=&INSDSN..&ROOT.&ROOT.PE(+1)
- Initial sequential image of error messages
PAC7LE: DSN=&INSDSN..&ROOT.&ROOT.AE0

.Transaction file:

- Update transactions
PAC7MC: DSN=&&PARMMB

.Permanent output files:

- Error messages to be rebuilt
PAC7AE: DSN=&INSDV..&ROOT.&ROOT.AE
- User parameters to be recreated
PAC7AP: DSN=&INSDV..&ROOT.&ROOT.AP

.Output report:

- Reconstruction report
PAC7IJ

.Sort file(s):

- SORTWK01
- SORTWK02
- SORTWK03

DATABASE MANAGEMENT UTILITIES
PARM: UPDATE OF USER PARAMETERS
PARM: DESCRIPTION OF STEPS

PAGE

127

2
10
9

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)

DATABASE MANAGEMENT UTILITIES

2

PARM: UPDATE OF USER PARAMETERS

10

PARM: EXECUTION JCL

10

2.10.10. PARM: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * - UPDATE OF USER PARAMETERS -
//*****
// $RADP.PARM PROC ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// *: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// SORTLIB='$BIBT', SORT LIBRARY
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK, WORK FILES UNIT
// CYL=1, SORTWORK SPACE
// VOLS='SER=$VOLUN', PARAMETERS BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// DSCB='$DSCB', DSCB MODEL FILE
// SPAPE='(TRK,(10,1),RLSE)' BACKUP SPACE
//*****
// 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=3440)
// VERIFY EXEC PGM=IDCAMS
//*****
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// SYSPRINT DD SYSOUT=&OUT
// PAC7AP DD DSN=&INDSV..&ROOT.&ROOT.AP,DISP=OLD
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=OLD
// SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAP),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
// PACU15 EXEC PGM=PACU15
//*****
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// PAC7DD DD SYSOUT=&OUT
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=OLD
// PAC7EC DD DSN=&INDSN..&ROOT.&ROOT.PE(0),DISP=SHR
// PAC7IJ DD SYSOUT=&OUT
// PAC7MC DD DSN=&&PARMMB,DISP=(OLD,PASS)
// PAC7AP DD DSN=&INDSV..&ROOT.&ROOT.AP,DISP=OLD
// PAC7CE DD DSN=&INDSN..&ROOT.&ROOT.PE(+1),DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPE,
// DCB=( &DSCB,RECFM=FB,LRECL=80,BLKSIZE=6160)
// SYSUDUMP DD SYSOUT=&OUT
// DEFINE EXEC PGM=IDCAMS,COND=(0,NE,PACU15)
//*****
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// SYSPRINT DD SYSOUT=&OUT
// SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AE),
// DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AP),
// DISP=SHR
// PACU80 EXEC PGM=PACU80,COND=(0,NE,PACU15),REGION=0K
//*****
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// SORTLIB DD DSN=&SORTLIB,DISP=SHR
// *: STEPCAT DD DSN=&SYSTCAT,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// SYSOUX DD SYSOUT=&OUT
// PAC7IJ DD SYSOUT=&OUT
// PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=OLD
// PAC7AP DD DSN=&INDSV..&ROOT.&ROOT.AP,DISP=OLD
// PAC7LE DD DSN=&INDSN..&ROOT.&ROOT.AE0,DISP=SHR
// PAC7MC DD DSN=&&PARMMB,DISP=(OLD,DELETE)

```


DATABASE MANAGEMENT UTILITIES

PARM: UPDATE OF USER PARAMETERS

PARM: EXECUTION JCL

2

10

10

```
//PAC7CE DD DSN=&INSDN..&ROOT.&ROOT.PE(+1),DISP=OLD
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//DELPE EXEC PGM=IEFBR14,COND=(8,NE,PACU15)
//*****
//DDPE DD DSN=&INSDN..&ROOT.&ROOT.PE(+1),DISP=(OLD,DELETE)
```

VisualAge Pacbase - Operation Manual
BATCH PROC.: ADMINISTRATOR'S GUIDE
VERSIONING UTILITIES

PAGE 130

3

3. VERSIONING UTILITIES

VERSIONING UTILITIES	PAGE	131
PEI: PRODUCTION ENVIRONMENT INTERFACE		3
		1

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.

	PAGE	132
VERSIONING UTILITIES		3
PEI: PRODUCTION ENVIRONMENT INTERFACE		1
PEI: OVERVIEW		1

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.

	PAGE	133
VERSIONING UTILITIES		3
PEI: PRODUCTION ENVIRONMENT INTERFACE		1
INPE: FILE INITIALIZATION		2

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.

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
INPE: FILE INITIALIZATION

PAGE

134

3
1
2

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=&INDUV..&ROOT.&FILE.AR
-Index file
 PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
-Error message file
 PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

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

.Output file:
-PEI initial backup
 PAC7PP: DSN=&INDUN..&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

VERSIONING UTILITIES

3

PEI: PRODUCTION ENVIRONMENT INTERFACE

1

INPE: FILE INITIALIZATION

2

3.1.2.3. INPE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * *
// * - PEI OPTION - BACKUP FILE INITIALIZATION - *
//*****
//$RADP.INPE PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUN='$INDUN', INDEX OF NON-VSAM USER FILES
// INDUV='$INDUV', INDEX OF USER VSAM FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// DSCB='$DSCB', DSCB MODEL FILE
// UWK=$UWK, WORK UNIT
// SORTLIB='$BIBT', SORT LIBRARY
// CYL=1, SORTWORK SPACE
// SPAPP='(TRK,(5,2),RLSE)', BACKUP SPACE
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&INPEMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PACR01 EXEC PGM=PACR01
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&INPEMB,DISP=(OLD,PASS)
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7IB DD SYSOUT=&OUT,DCB=BLKSIZE=101
//PAC7PP DD DSN=&INDUN..&ROOT.&FILE.PP(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPP,
// DCB=(&DSCB,LRECL=110,RECFM=FB,BLKSIZE=6160)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

VERSIONING UTILITIES	PAGE	136
PEI: PRODUCTION ENVIRONMENT INTERFACE		3
SVPE: FILE BACKUP		1
		3

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.

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
SVPE: FILE BACKUP

PAGE

137

3
1
3

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=&INDUV..&ROOT.&FILE.AB
- 'On-line' PEI file
PAC7AC: DSN=&INDUV..&ROOT.&FILE.AC
- Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
- Error message file
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

.Output file:
- PEI backup
PAC7PP: DSN=&INDUN..&ROOT.&ROOT.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

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

SVPE: FILE BACKUP

PAGE

138

3

1

3

BACKUP CANCELLATION IN CASE OF INCOHERENCE: IEFBR14

.Input file:

-PEI Backup

DDPP: DSN=&INDUN.&ROOT.&FILE.PP(+1)

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

VERSIONING UTILITIES

3

PEI: PRODUCTION ENVIRONMENT INTERFACE

1

SVPE: FILE BACKUP

3

3.1.3.3. SVPE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * *
// * - PEI OPTION : FILES BACKUP - *
//*****
//$RADP.SVPE PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUN='$INDUN', INDEX OF NON-VSAM USER FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK, WORK UNIT
// VOLS='SER=$VOLUN', BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// DSCB='$DSCB', DSCB MODEL FILE
// SPAPP='(TRK,(5,2),RLSE)' BACKUP SPACE
//*****
//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=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PACR60 EXEC PGM=PACR60
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7IE DD SYSOUT=&OUT,DCB=BLKSIZE=81
//PAC7MB DD DSN=&&SVPEMB,DISP=(OLD,DELETE)
//PAC7PP DD DSN=&INDUN..&ROOT.&FILE.PP(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPP,
// DCB=(&DSCB,LRECL=110,RECFM=FB,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT
//DELPP EXEC PGM=IEFBR14,COND=(8,NE,PACR60)
//*****
//DDPP DD DSN=&INDUN..&ROOT.&FILE.PP(+1),DISP=(OLD,DELETE)

```

VERSIONING UTILITIES	PAGE	140
PEI: PRODUCTION ENVIRONMENT INTERFACE		3
RSPE: FILE RESTORATION		1
		4

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.

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
RSPE: FILE RESTORATION

PAGE 141
3
1
4

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=&INDSV..&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=&INDSV..&ROOT.&ROOT.AE
-Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
-PEI backup file
PAC7PP: DSN=&INDUN..&ROOT.&ROOT.PP(0)

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

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

RSPE: FILE RESTORATION

PAGE

142

3

1

4

.Output reports:

-Review

PAC7IF

-Batch-procedure authorization option

PAC7DD

.Return code:

8: No authorization on batch procedures

VERSIONING UTILITIES

3

PEI: PRODUCTION ENVIRONMENT INTERFACE

1

RSPE: FILE RESTORATION

4

3.1.4.3. RSPE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * - PEI OPTION : FILES RESTORATION -
//*****
//$RADP.RSPE PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUN='$INDUN', INDEX OF NON-VSAM USER FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// UWK=$UWK, WORK UNIT
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT OUTPUT CLASS
//*****
//INPUT EXEC PGM=PTU004
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPSCAT DD DSN=&SYSTCAT,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7AE DD DSN=&INDSV. &ROOT.&ROOT.AE,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&MB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//DELDEF EXEC PGM=IDCAMS,COND=(0,NE,INPUT)
//*****
//*:STEPSCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AR DD DSN=&INDUV. &ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDUN. &ROOT.&FILE.SY(DF&ROOT.&FILE.AB),DISP=SHR
// DD DSN=&INDUN. &ROOT.&FILE.SY(DF&ROOT.&FILE.AC),DISP=SHR
// DD DSN=&INDSN. &ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PACR61 EXEC PGM=PACR61,COND=(0,NE,INPUT)
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPSCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&MB,DISP=(OLD,DELETE)
//PAC7AB DD DSN=&INDUV. &ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV. &ROOT.&FILE.AC,DISP=SHR
//PAC7AR DD DSN=&INDUV. &ROOT.&FILE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV. &ROOT.&ROOT.AE,DISP=SHR
//PAC7IF DD SYSOUT=&OUT,DCB=BLKSIZE=81
//PAC7PP DD DSN=&INDUN. &ROOT.&FILE.PP(0),DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	144
VERSIONING UTILITIES		3
PEI: PRODUCTION ENVIRONMENT INTERFACE		1
PRPE: PRODUCTION ENVIRONMENT PRINTOUTS		5

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 resolved, the PRPE procedure can be restarted as it is.

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
PRPE: PRODUCTION ENVIRONMENT PRINTOUTS

PAGE

145

3
1
5

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.

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
PRPE: PRODUCTION ENVIRONMENT PRINTOUTS

PAGE

146

3
1
5

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=&INDUV..&ROOT.&FILE.AB
- 'On-line' PEI file
PAC7AC: DSN=&INDUV..&ROOT.&FILE.AC
- Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
- Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
- Error-message file
PAC7AE: DSN=&INDSV..&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

VERSIONING UTILITIES

3

PEI: PRODUCTION ENVIRONMENT INTERFACE

1

PRPE: PRODUCTION ENVIRONMENT PRINTOUTS

5

3.1.5.4. PRPE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * - PEI OPTION - PRODUCTION ENVIRONMENT PRINTING -
//*****
//$RADP.PRPE PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK, WORK UNIT
// SORTLIB='$BIBT', SORT LIBRARY
// CYL=1 SORTWORK SPACE
//*****
// * INPUT : PRINTING REQUEST
// * COL 2-3 : 'PL' (CARD CODE)
// * COL 4 : '1' ENVIRONMENTS PER LIBRARY
// * COL 5 : '1' LIBRARIES PER ENVIRONMENT
// * COL 6 : '1' ENTITIES PER ENVIRONMENT
// * COL 7 : '1' ENTITIES PER SESSION
// * COL 8 : '1' ENVIRONMENTS PER VA PAC ENTITY
// * COL 9 : '1' ENVIRONMENTS PER ENTITY (EXT.NAME)
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&PRPEMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAM5
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PACR10 EXEC PGM=PACR10
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7IE DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB DD DSN=&&PRPEMB,DISP=(OLD,DELETE)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUX DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	148
VERSIONING UTILITIES		3
PEI: PRODUCTION ENVIRONMENT INTERFACE		1
GRPE: TRANSACTION-GENERATION FOR REORGANIZATION		6

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.

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
GRPE: TRANSACTION-GENERATION FOR REORGANIZATION

PAGE

149

3
1
6

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=&INDUV..&ROOT.&FILE.AB
- 'On-line' PEI file
PAC7AC: DSN=&INDUV..&ROOT.&FILE.AC
- Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
- Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
- Error message file
PAC7AE: DSN=&INDSV..&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.

VERSIONING UTILITIES

3

PEI: PRODUCTION ENVIRONMENT INTERFACE

1

GRPE: TRANSACTION-GENERATION FOR REORGANIZATION

6

3.1.6.3. GRPE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// *          - PEI OPTION  GENERATION OF TRANSACTIONS FOR -
// *          REORGANIZING THE PACBASE DATABASE
//*****
//$RADP.GRPE PROC FILE=$FILE,          NUMBER OF PHYSICAL DATABASE
//          ROOT=$ROOT,                ROOT OF THE VA PAC SYSTEM
//          INDSV=' $INDSV',            INDEX OF VSAM SYSTEM FILES
//          INDSN=' $INDSN',            INDEX OF NON-VSAM SYSTEM FILES
//          INDUV=' $INDUV',            INDEX OF VSAM USER FILES
//*:          VSAMCAT=' $VCAT',          USER VSAM CATALOG
//*:          SYSTCAT=' $SCAT',          VA PAC SYSTEM VSAM CATALOG
//          STEPLIB=' $MODB',            LIBRARY OF LOAD-MODULES
//          OUT=$OUT,                   OUTPUT CLASS
//          UWK=$UWK,                   WORK UNIT
//          SORTLIB=' $BIBT',           SORT LIBRARY
//          CYL=1                        SORTWORK SPACE
//*****
//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=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
//          DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
//          DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PACR40 EXEC PGM=PACR40
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&GRPEMB,DISP=(OLD,DELETE)
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7IK DD SYSOUT=&OUT,DCB=BLKSIZE=101
//PAC7MV DD DSN=&&GRPEREOR,DISP=(,PASS),
//          UNIT=&UWK,
//          SPACE=(TRK,(10,1),RLSE),
//          DCB=(LRECL=80,RECFM=FB,BLKSIZE=3440)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUX DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	151
VERSIONING UTILITIES		3
PEI: PRODUCTION ENVIRONMENT INTERFACE		1
HIPE: AUTOMATIC SESSION FREEZE		7

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.

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

HIPE: AUTOMATIC SESSION FREEZE

3

1

7

3.1.7.2. HIPE: USER INPUT

HIPE: USER INPUT

A required '*' line:

```

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

```

An optional session freeze line:

```

-----
!POS.!LEN.! VALUE ! MEANING !
!-----!
! 2 ! 2 ! ! ! Line code !
! ! ! 'X1' ! if the entities have been put into !
! ! ! ! production !
! ! ! 'X4' ! if no entity has been put into produc- !
! ! ! ! tion !
! 4 ! 4 ! 'HIST' ! Freeze request !
! 8 ! 60 ! ! Freeze comments !
! 68 ! 4 ! ssss ! Forcing of session number (number com- !
! ! ! ! prised 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.

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
HIPE: AUTOMATIC SESSION FREEZE

PAGE

153

3
1
7

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=&INDUV..&ROOT.&FILE.AR
-Error message file
PAC7AE: DSN=&INDSV..&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=&INDUV..&ROOT.&FILE.AB
-'On-line' PEI file
PAC7AC: DSN=&INDUV..&ROOT.&FILE.AC
-Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
-Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
-Journal file
PAC7AJ: DSN=&INDUV..&ROOT.&FILE.AJ
-Error message file
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE
HIPE: AUTOMATIC SESSION FREEZE

PAGE

154

3
1
7

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

.Output report:
-Execution report
PAC7IG

.Work files:
PAC7MW
PAC7WB

VERSIONING UTILITIES

3

PEI: PRODUCTION ENVIRONMENT INTERFACE

1

HIPE: AUTOMATIC SESSION FREEZE

7

3.1.7.4. HIPE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * - PEI OPTION - AUTOMATIC SESSION FREEZE -
//*****
//$RADP.HIPE PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK WORK UNIT
//*****
// * INPUT : REQUESTED ' * ' LINE
// * OPTIONAL 'X1' LINE (AUTOMATICALLY GENERATED)
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&HIPEMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PTUBAS EXEC PGM=PTUBAS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7DS DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACR30 EXEC PGM=PACR30,COND=(0,NE,PTUBAS)
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV..&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV..&ROOT.&FILE.AC,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&ROOT.&FILE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7IG DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB DD DSN=&&HIPEMB,DISP=(OLD,DELETE)
//PAC7MW DD UNIT=&UWK,
// SPACE=(TRK,(1,1),RLSE),
// DCB=(LRECL=167,RECFM=FB,BLKSIZE=6179)
//PAC7WB DD UNIT=&UWK,
// SPACE=(TRK,(2,1),RLSE),
// DCB=(LRECL=110,RECFM=FB,BLKSIZE=6160)

```

VERSIONING UTILITIES	PAGE	156
PEI: PRODUCTION ENVIRONMENT INTERFACE		3
SIPE: PRODUCTION TURNOVER SIMULATION		1
		8

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

Once the problem is resolved, the SIPE procedure can be restarted again just as before.

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

SIPE: PRODUCTION TURNOVER SIMULATION

3

1

8

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)
-----

```

An 'EE' line identifying the environment is 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
-----

```

An '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 enviro-
!   !   !         ! nment if different from code in
!   !   !         ! Database
! 20 !  8 !nnnnnnnn! Entity external name in source enviro-
!   !   !         ! nment if transfer with RENAME
-----

```

VERSIONING UTILITIES
PEI: PRODUCTION ENVIRONMENT INTERFACE
SIPE: PRODUCTION TURNOVER SIMULATION

PAGE

158

3
1
8

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=&INDUV..&ROOT.&FILE.AB
- 'On-line' PEI file
PAC7AC: DSN=&INDUV..&ROOT.&FILE.AC
- Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
- Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
- Error message file
PAC7AE: DSN=&INDSV..&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

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

SIPE: PRODUCTION TURNOVER SIMULATION

3

1

8

3.1.8.4. SIPE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * - PEI OPTION - SIMULATION -
//*****
//$RADP.SIPE PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN='$INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// UWK=$UWK, WORK UNIT
// OUT=$OUT OUTPUT CLASS
//*****
// * INPUT : REQUESTED '*' LINE
// * REQUESTED 'EE' LINE
// * ONE 'EU' LINE PER ENTITY
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&SIPEMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV.&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV.&ROOT.&FILE.AC,DISP=SHR
//PAC7AN DD DSN=&INDUV.&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV.&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN.&ROOT.&ROOT.SY(VERIFAB),DISP=SHR
// DD DSN=&INDSN.&ROOT.&ROOT.SY(VERIFAC),DISP=SHR
// DD DSN=&INDSN.&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN.&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PACR22 EXEC PGM=PACR22
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7AB DD DSN=&INDUV.&ROOT.&FILE.AB,DISP=SHR
//PAC7AC DD DSN=&INDUV.&ROOT.&FILE.AC,DISP=SHR
//PAC7AE DD DSN=&INDSV.&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV.&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV.&ROOT.&FILE.AR,DISP=SHR
//PAC7IE DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB DD DSN=&&SIPEMB,DISP=(OLD,DELETE)
//PAC7MT DD DSN=&&PAC7MT,DISP=(,PASS),UNIT=&UWK,
// SPACE=(TRK,(1,1),RLSE),
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=3440)

```

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 more other sessions.

Pac/transfer scans the archived Journal file and consults a dedicated Parameters file.

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

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

Then, Pac/transfer generates transactions for the target environment(s) also defined by these parameters.

These transactions are used by the VA Pac batch updating 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 more target sessions.

When a development is finished in a test session it is possible to transfer this session's contents into another session specific to validation, and if necessary simultaneously to a second session for production turnover.

In the transfer file, the selected transactions of the source session are duplicated as often as there are target sessions.

There are no constraints regarding the ordering of sessions. It is possible to transfer the state of a source session to a later target session (the target session number greater than that of the source session), just as it is possible to transfer it to a previous target session (the 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.

	PAGE	162
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRUP: TRANSFER-PARAMETER UPDATE		1

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.

	PAGE	163
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRUP: TRANSFER-PARAMETER UPDATE		1

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 explicitly.

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.

	PAGE	164
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRUP: TRANSFER-PARAMETER UPDATE		1

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.

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRUP: TRANSFER-PARAMETER UPDATE

1

3.2.1.2. TRUP: USER INPUT

TRUP: USER INPUT

. User identification line (required)

```

-----
!Pos.! Len.! Value  ! Meaning
-----
!  2 !   1 ! '*'    ! Line code
!  3 !   8 ! uuuuuuu ! 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)
-----

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRUP: TRANSFER-PARAMETER UPDATE

1

. 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 !
-----

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRUP: TRANSFER-PARAMETER UPDATE

1

. DSMS-change 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 ! '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                                         !
-----

```

VERSIONING UTILITIES
PAC/TRANSFER
TRUP: TRANSFER-PARAMETER UPDATE

PAGE

168

3
2
1

. 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 !lllll ! 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 !  
-----
```


VERSIONING UTILITIES
PAC/TRANSFER
TRUP: TRANSFER-PARAMETER UPDATE

PAGE

169

3
2
1

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
```

VERSIONING UTILITIES

PAC/TRANSFER

TRUP: TRANSFER-PARAMETER UPDATE

PAGE

170

3

2

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

CLot1 GCPR 001220 PR 001250

CLot1 GCPR 001220001JEAN

1

VERSIONING UTILITIES
PAC/TRANSFER
TRUP: TRANSFER-PARAMETER UPDATE

PAGE 171
3
2
1

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=&INDUV..&ROOT.&FILE.AR
-Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
-Error-message file
PAC7AE: DSN=&INDSV..&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=&INDUV..&ROOT.&FILE.UV

.Work file:

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

.Output reports:

-Input check
PAC7ET
-User check
PAC7DD

VERSIONING UTILITIES

PAC/TRANSFER

TRUP: TRANSFER-PARAMETER UPDATE

PAGE

172

3

2

1

SELECTION-PARAMETER PRINTOUT: PTUG11

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

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRUP: TRANSFER-PARAMETER UPDATE

1

3.2.1.4. TRUP: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * *
// * PAC/TRANSFER: *
// * UPDATE OF THE TRANSFER PARAMETERS *
//*****
//$RADP.TRUP PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDUV=' $INDUV', INDEX OF VSAM USER FILE
// INDSV=' $INDSV', INDEX OF VSAM SYSTEM FILE
// OUT=$OUT, OUTPUT CLASS
//*: VSAMCAT=' <>', USER VSAM CATALOG
//*: SYSTCAT=' <>', SYSTEM VSAM CATALOG
// STEPLIB=' $MODB', LOAD-MODULES LIBRARY
// SORTLIB=' $BIBT', SORT LIBRARY
// CYL=5, SORTWORK SPACE
// UWK=$UWK WORK UNIT
//*****
//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=PTUG10
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7UV DD DSN=&INDUV..&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=PTUG11,COND=(8,LE,PTUG10)
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV DD DSN=&INDUV..&ROOT.&FILE.UV,DISP=SHR
//PAC7GL DD DSN=&TRUPGL,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=19,BLKSIZE=1900)
//PAC7ET DD SYSOUT=&OUT
//PTUG12 EXEC PGM=PTUG12,COND=(8,LE,PTUG10)
//SYSUDUMP DD SYSOUT=&OUT
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV DD DSN=&INDUV..&ROOT.&FILE.UV,DISP=SHR

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRUP: TRANSFER-PARAMETER UPDATE

1

```
//PAC7GL DD DSN=&&TRUPGL,DISP=(OLD,DELETE)
//PAC7ML DD DSN=&&TRUPML,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
/*
```

	PAGE	175
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRJC: COMPRESSION OF ARCHIVED JOURNAL		2

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 that only comprises effective transactions, 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.

	PAGE	176
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRJC: COMPRESSION OF ARCHIVED JOURNAL		2

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 !	! uuuuuuu	! User code
! 11 !	! 8 !	! ppppppp	! 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

VERSIONING UTILITIES
PAC/TRANSFER
TRJC: COMPRESSION OF ARCHIVED JOURNAL

PAGE

178

3
2
2

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=&INDUN..&ROOT.&FILE.PJ
-Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN
-Error-message file
PAC7AE: DSN=&INDSV..&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

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRJC: COMPRESSION OF ARCHIVED JOURNAL

2

.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=&INDUV..&ROOT.&FILE.AN

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

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

.Sort file(s):
SORTWK01
SORTWK02
SORTWK03

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRJC: COMPRESSION OF ARCHIVED JOURNAL

2

3.2.2.4. TRJC: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * PAC/TRANSFER -
// * COMPRESSION OF ARCHIVED JOURNAL
//*****
//$RADP.TRJC PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF VA PAC SYSTEM
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDUN='$INDUN', INDEX OF NON-VSAM USER FILES
// SPAJT='(TRK,(60,15),RLSE)', COMPRESSED JOURNAL SPACE
// SPAPJ='(TRK,(150,15),RLSE)', TEMPORARY FILE SPACE
// UNITS=$UNITUN, COMPRESSED JOURNAL FILE
// VOLS='$SER=$VOLUN', COMPRESSED JOURNAL FILE VOLUME
// OUT=$OUT, OUTPUT CLASS
//*: VSAMCAT='<>', VSAM USER CATALOG
//*: SYSTCAT='<>', VSAM SYSTEM CATALOG
// STEPLIB='$MODB', LOAD-MODULES LIBRARY
// SORTLIB='$BIBT', SORT LIBRARY
// CYL=(10,2), SORTWORK SIZE
// UWK=$UWK, WORK UNIT
//*****
//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)
//DELJT EXEC PGM=IDCAMS
//*****
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&ROOT.&FILE.SY(DL&ROOT.&FILE.JT),
// DISP=SHR
//PTUG05 EXEC PGM=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)
//*: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7MB DD DSN=&&TRJCMB,DISP=(OLD,DELETE)
//PAC7PJ DD DSN=&INDUN..&ROOT.&FILE.PJ(0),DISP=OLD
//PAC7GP DD DSN=&&PAC7GP,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=223,BLKSIZE=26760),
// SPACE=&SPAPJ
//PAC7ET DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//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=&SPAPJ
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRJC: COMPRESSION OF ARCHIVED JOURNAL

2

```
//*
//PTUG07 EXEC PGM=PTUG07,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)
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7PK DD DSN=&&PAC7PK,DISP=(OLD,PASS)
//PAC7PL DD DSN=&INDUN..&ROOT.&FILE.JT,
//      DISP=(,CATLG),
//      UNIT=&UNITS,VOL=&VOLS,
//      DCB=(RECFM=FB,LRECL=167,BLKSIZE=6179),
//      SPACE=&SPAJT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
```

	PAGE	182
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRPF: TRANSFER-FILE CREATION		3

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.

VERSIONING UTILITIES
 PAC/TRANSFER
 TRPF: TRANSFER-FILE CREATION

3
 2
 3

3.2.3.2. TRPF: USER INPUT

TRPF: USER INPUT

. User identification line (required)

```
-----
!Pos.! Len.! Value  ! Meaning  !
!-----+-----+-----!
!  2 !   1 ! '*'    ! Line code !
!  3 !   8 ! uuuuuuu ! User code !
! 11 !   8 ! ppppppp ! Password  !
-----
```

. Transaction Set for processing selection line (required)

```
-----
!Pos.! Len.! Value  ! Meaning  !
!-----+-----+-----!
!  2 !   2 ! 'LT'   !          !
!  4 !   5 ! lllll  ! 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).

VERSIONING UTILITIES
PAC/TRANSFER
TRPF: TRANSFER-FILE CREATION

PAGE

184

3
2
3

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=&INDUV..&ROOT.&FILE.AR
-Error-message file
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
-Parameter file
PAC7UV: DSN=&INDUV..&ROOT.&FILE.UV
-Sequential or compressed file
PAC7JT: DSN=&INDUN..&ROOT.&FILE.JT

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

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

.Sort file(s):
SORTWK01
SORTWK02
SORTWK03

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

VERSIONING UTILITIES
 PAC/TRANSFER
 TRPF: TRANSFER-FILE CREATION

3
 2
 3

3.2.3.4. TRPF: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * PAC/TRANSFER -
// * CREATING THE TRANSFER FILE
//*****
//$RADP.TRPF PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
// INDUN='$INDUN', INDEX OF USER NON-VSAM FILE (TJ)
// SPATJ='(TRK,(60,15),RLSE)', JOURNAL TO REPORT
// UNITS=$UNITUN, NON-VSAM USER FILE (TJ)
// VOLS='SER=$VOLUN', NUMBER OF NON-VSAM USER FILE (TJ)
// OUT=$OUT, OUTPUT CLASS
//*: VSAMCAT='<>', VSAM USER CATALOG
//*: SYSTCAT='<>', VSAM SYSTEM CATALOG
// STEPLIB='$MODB', LOAD-MODULES LIBRARY
// SORTLIB='$BIBT', SORT LIBRARY
// CYL=5, SORTWORK SPACE
// UWK=$UWK WORK UNIT
//*****
//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=&INDUN..&ROOT.&FILE.SY(DL&ROOT.&FILE.TJ),
// DISP=SHR
//PTUG50 EXEC PGM=PTUG50
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//PAC7MB DD DSN=&&TRPFMB,DISP=(OLD,DELETE)
//PAC7UV DD DSN=&INDUV..&ROOT.&FILE.UV,DISP=SHR
//PAC7ER DD SYSOUT=&OUT
//PAC7ET DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7JT DD DSN=&INDUN..&ROOT.&FILE.JT,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7TJ DD DSN=&INDUN..&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

```

	PAGE	186
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRDU: DSMS-ENVIRONMENT PREPARATION		4

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.

	PAGE	187
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRDU: DSMS-ENVIRONMENT PREPARATION		4

This update must be executed BEFORE the transfer operation.

	PAGE	188
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRDU: DSMS-ENVIRONMENT PREPARATION		4

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.

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRDU: DSMS-ENVIRONMENT PREPARATION

4

3.2.4.2. TRDU: USER INPUT

TRDU: USER INPUT

. User identification line (required)

```

-----
!Pos.! Len.! Value  ! Meaning  !
!-----+-----+-----+-----!
!  2 !   1 ! '*'    ! Line code !
!  3 !   8 ! uuuuuuu ! User code  !
! 11 !   8 ! ppppppp ! Password  !
-----

```

. TRANSACTION SET selection line (required)

```

-----
!Pos.! Len.! Value  ! Meaning  !
!-----+-----+-----+-----!
!  2 !   2 ! 'LT'   !          !
!  4 !   5 ! lllll  ! Selected Transaction Set code !
!   !   ! '*****' ! Selection of all Sets        !
-----

```

One and only one LT-type line is required.

VERSIONING UTILITIES	
PAC/TRANSFER	
TRDU: DSMS-ENVIRONMENT PREPARATION	

PAGE	190
	3
	2
	4

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=&INDUV..&ROOT.&FILE.AR
-Error-messages file
  PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
-Parameter file
  PAC7UV: DSN=&INDUV..&ROOT.&FILE.UV
-User input
  PAC7MB: DSN=&&TRDUMB

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

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

PREPARATION OF DSMS BEFORE TRANSFERS: PTUG44

```
.Input files:
-Parameter file
  PAC7UV: DSN=&INDUV..&ROOT.&FILE.UV
-Error-message file
  PAC7AE: DSN=&INDSV.&ROOT.&ROOT.AE
-Data file
  PAC7AR: DSN=&INDSV.&ROOT.&FILE.AR
-VisualAge Pacbase element file
  PACDDC: DSN=&DSMS
-Batch-transaction file
  PAC7MB: DSN=&&TRDUMB
```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRDU: DSMS-ENVIRONMENT PREPARATION

4

.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=&INDSV.&ROOT.&ROOT.AE
-Data file
PAC7AR: DSN=&INDUV.&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

VERSIONING UTILITIES

PAC/TRANSFER

TRDU: DSMS-ENVIRONMENT PREPARATION

PAGE

192

3

2

4

CREATION OF BEFORE-TRANSFER TRANSACTIONS: IEBCGENER

.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: IEBCGENER

.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

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRDU: DSMS-ENVIRONMENT PREPARATION

4

3.2.4.4. TRDU: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * *
// * PAC/TRANSFER - *
// * PREPARING THE DSMS ENVIRONMENT *
//*****
//$RADP.TRDU PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDUV=' $INDUV', INDEX OF USER VSAM FILES
// INDSV=' $INDSV', INDEX OF SYSTEM FILES
// DSMS=' $DSMS', DSN OF DSMS VA PAC ELEMENTS FILE
// SPACI=' (TRK,(90,15),RLSE)', 'CI' FILE SPACE
// SPACC=' (TRK,(90,15),RLSE)', 'CC' FILE SPACE
// SPAGC=' (TRK,(90,15),RLSE)', 'GC' FILE SPACE
// SPAAV=' (TRK,(90,15),RLSE)', 'AV' FILE SPACE
// SPAAP=' (TRK,(90,15),RLSE)', 'AP' FILE SPACE
// CYL=2, NUMBER OF SORTWORK CYLINDERS
// OUT=$OUT, OUTPUT CLASS
// *: VSAMCAT=' <>', VSAM USER CATALOG
// *: SYSTCAT=' <>', VSAM SYSTEM CATALOG
// STEPLIB=' $MODB', LOAD-MODULES LIBRARY
// UWK=$UWK WORK UNIT
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&TRDUMB,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE)
// *
//PTUG42 EXEC PGM=PTUG42
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
//PAC7MB DD DSN=&&TRDUMB,DISP=(OLD,DELETE)
//PAC7UV DD DSN=&INDUV. &ROOT. &FILE.UV,DISP=SHR
//PAC7AE DD DSN=&INDSV. &ROOT. &ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV. &ROOT. &FILE.AR,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7ET DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7BM DD DSN=&&TRDUBM,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE)
//PTUG44 EXEC PGM=PTUG44
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
//PAC7MB DD DSN=&&TRDUBM,DISP=SHR
//PAC7UV DD DSN=&INDUV. &ROOT. &FILE.UV,DISP=SHR
//PAC7AE DD DSN=&INDSV. &ROOT. &ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV. &ROOT. &FILE.AR,DISP=SHR
//PACDDC DD DSN=&DSMS,DISP=SHR
//PAC7ET DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7CI DD DSN=&&TRDUCI,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=27500),
// UNIT=&UWK,SPACE=&SPACI
//PAC7SI DD DSN=&&TRDUSI,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=27500),
// UNIT=&UWK,SPACE=&SPACI
//PAC7GC DD DSN=&&TRDUGC,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=40,BLKSIZE=32000),
// UNIT=&UWK,SPACE=&SPAGC
//PTUG46 EXEC PGM=PTUG46
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRDU: DSMS-ENVIRONMENT PREPARATION

4

```

//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//PAC7AE DD DSN=&INDSV. &ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV. &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
//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)

```

	PAGE	195
VERSIONING UTILITIES		3
PAC/TRANSFER		2
UPDATE OF DSMS FUNCTION BEFORE VA PAC UPDATE		5

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.

VERSIONING UTILITIES	PAGE	196
PAC/TRANSFER		3
TRRP: GENERATION OF TRANSFER TRANSACTIONS		2
		6

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.

	PAGE	197
VERSIONING UTILITIES		3
PAC/TRANSFER		2
TRRP: GENERATION OF TRANSFER TRANSACTIONS		6

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).

VERSIONING UTILITIES	PAGE	198
PAC/TRANSFER		3
TRRP: GENERATION OF TRANSFER TRANSACTIONS		2
		6

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 ! uuuuuuu ! 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         !  
!   !   ! '*****' ! . 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 !  
-----
```

VERSIONING UTILITIES	
PAC/TRANSFER	
TRRP: GENERATION OF TRANSFER TRANSACTIONS	

PAGE	200
	3
	2
	6

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=&INDUV.&ROOT.&FILE.AR
-Error messages
 PAC7AE : DSN=&INDSV.&ROOT.&ROOT.AE
-Parameter-setting file
 PAC7UV : DSN=&INDUV.&ROOT.&FILE.UV
-Compressed journal file
 PAC7JT : DSN=&INDUN.&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

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRRP: GENERATION OF TRANSFER TRANSACTIONS

6

EXTRACTION: PACX

This step extracts transactions based on user input.

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

VERSIONING UTILITIES

PAGE

202

PAC/TRANSFER

3

2

TRRP: GENERATION OF TRANSFER TRANSACTIONS

6

.Return code(s):
0: No error
8: Serious error (detailed in PAC7DD)

POSITIONING THE 'PROCESSED SESSION' ATTRIBUTE: PTUG61

.Permanent input files:
-Index file
PAC7AR: DSN=&INDUV..&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=&INDUV..&ROOT.&FILE.UV

.Output report(s):
-Transfer statistics
PAC7ET

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRRP: GENERATION OF TRANSFER TRANSACTIONS

6

3.2.6.4. TRRP: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * *
// * PAC/TRANSFER - *
// * GENERATING THE TRANSFER TRANSACTIONS *
//*****
//$RADP.TR RP PROC FILE=$FILE, NUMBER OF THE PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDUV=' $INDUV', INDEX OF VSAM USER FILES
// INDSV=' $INDSV', INDEX OF VSAM SYSTEM FILES
// INDSN=' $INDSN', INDEX OF NON-VSAM SYSTEM FILES
// INDUN=' $INDUN', INDEX OF NON-VSAM USER FILES
// OUT=$OUT, OUTPUT CLASS
//*: VSAMCAT=' <>', VSAM USER CATALOG
//*: SYSTCAT=' <>', VSAM SYSTEM CATALOG
// STEPLIB=' $MODB', LOAD-MODULES LIBRARY
// SORTLIB=' $BIBT', SORT LIBRARY
// CYL=5, SORTWORK SPACE
// UWK=$UWK, WORK UNIT
// SPAMV=' (TRK,(30,10))', EXTRACTED TRANSACTIONS SPACE
// SPAPJ=' (TRK,(30,10))', SELECTED PJ FILE SPACE
// SPAMB=' (TRK,(5,1),RLSE)', SPACE OF COMMAND FILE
// SPATD=' (TRK,(50,10),RLSE)', SPACE OF CPSN FILE
//*****
//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
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PTUG60 EXEC PGM=PTUG60
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7UV DD DSN=&INDUV..&ROOT.&FILE.UV,DISP=SHR
//PAC7JT DD DSN=&INDUN..&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)
//PAC7DD DD SYSOUT=&OUT
//PAC7ET 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=&INDSN..&ROOT.&ROOT.SY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(REPRO999),DISP=SHR
//PACX EXEC PGM=PACX,REGION=0K
//*****
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRRP: GENERATION OF TRANSFER TRANSACTIONS

6

```

//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV. &ROOT. &ROOT. AE,DISP=SHR
//PAC7AN DD DSN=&INDUV. &ROOT. &FILE. AN,DISP=SHR
//PAC7AR DD DSN=&INDUV. &ROOT. &FILE. AR,DISP=SHR
//PAC7PJ DD DSN=&&PACXPJ,DISP=SHR
//PAC7IA DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7EE DD SYSOUT=&OUT
//PAC7EP DD SYSOUT=&OUT
//PAC7EQ DD SYSOUT=&OUT
//PAC7EZ DD SYSOUT=&OUT
//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)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//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=PTUG61,COND=((0,NE,PACX),(0,NE,PTUG60))
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV. &ROOT. &ROOT. AE,DISP=SHR
//PAC7AR DD DSN=&INDUV. &ROOT. &FILE. AR,DISP=SHR
//PAC7UV DD DSN=&INDUV. &ROOT. &FILE. UV,DISP=SHR
//PAC7MB DD DSN=&&PACXMB,DISP=(OLD,DELETE)
//PAC7ET DD SYSOUT=&OUT

```

	PAGE	205
VERSIONING UTILITIES		3
PAC/TRANSFER		2
UPDATE OF THE VISUALAGE PACBASE DATABASE		7

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).

	PAGE	206
VERSIONING UTILITIES		3
PAC/TRANSFER		2
REINITIALIZATION OF THE DSMS ENVIRONMENT		8

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).

3.3. TEAMCONNECTION

INTRODUCTION

TeamConnection is an integrated configuration management product which manages source programs, load modules, JCL, etc. It provides control functions and development guidelines for applications, and for their operational implementation.

The VA Pac/TeamConnection Interface allows you to integrate VA Pac-generated objects into the TeamConnection management environment.

The Interface enables the user to know:

- In VA Pac, the TeamConnection 'target' contexts where the generated objects are managed, i.e.: Family, Component, Release, and WorkArea;
- In TeamConnection, the 'source' context of these objects in VA Pac: Library Code, Session Number, User Code, Generation Date and Time. The 'source' context is defined in the 'Configurable Fields' specific to the Interface.

The Interface performs two main actions:

- 1- Generation of VisualAge Pacbase entities, followed by an import of generated objects into TeamConnection.

This step chains the execution of two procedures: the GPRT generation procedure, and the TCGP procedure, which creates the TeamConnection import actions for the generated object. The settings of the before/after cards for the object must be correct. (Please refer to the VA Pac-TeamConnection Interface Reference Manual).

- 2- Update of the TeamConnection contexts of the objects generated in VisualAge Pacbase.

First, in the TeamConnection environment, execute the PTC_CITC procedure, which extracts a list of VisualAge Pacbase objects.

In the VisualAge Pacbase environment, execute the TCCI procedure, which matches the extracted list with the list in the VA Pac Dictionary. This procedure writes a file containing the transactions to be updated in VisualAge Pacbase via the UPDT procedure.

	PAGE	208
VERSIONING UTILITIES		3
TEAMCONNECTION		3
TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION		1

3.3.1. TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION
3.3.1.1. TCGP: INTRODUCTION

TCGP: INTRODUCTION

The TCGP procedure completes the file produced by the GPRT VA-Pacbase Generation-Print procedure, and prepares the operations for importing generated objects in Team-Connection

CONDITION D'EXECUTION

The GPRT must be run first.

VERSIONING UTILITIES	
TEAMCONNECTION	
TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION	

PAGE	209
	3
	3
	1

3.3.1.2. TCGP: DESCRIPTION OF STEPS

PREPARATION OF IMPORT ACTIONS FOR TEAMC: PTC100

```
.Permanent input files:
-Target LIBRARY and SESSION file
  PAC7TS: DSN=&INDSV..&ROOT.&ROOT.TS
-Error messages
  PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
-Data file
  PAC7AR: DSN=&INDSV..&ROOT.&FILE.AR
-Index file
  PAC7AR: DSN=&INDSV..&ROOT.&FILE.AN

.Input file:
-VA Pac generated code:
  PAC7JB: DSN=&&PAC7JB

.Work file:
-PAF standard KSDS file
  SYSPAF: DSN=&SYSPAF..&USER

.Output report:
-Execution-related errors
  PAC7ET

.Output file:
-File for import in TeamConnection, to be submitted
  for execution
  PAC7BJ: DSN=&&PAC7BJ
```

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION

1

3.3.1.3. TCGP: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * -- TEAMCONNECTION - SETTING VALUES TO PARAMETERS -- *****
//*****
//$RADP.TCGP PROC FILE=$FILE, VA PAC PHYSICAL-DATABASE NUMBER
// ROOT=$ROOT, SYSTEM ROOT
// INDSV='$INDSV', VSAM SYSTEM FILES INDEX
// INDSN='$INDSN', NON-VSAM SYSTEM FILES INDEX
// INDUV='$INDUV', VSAM USER FILES INDEX
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LOAD-MODULES LIBRARY
// OUT=$OUT, PRINTOUT CLASS
// UWK=$UWK, WORK UNIT
// PAC7GB='????', 'PAC7GB' DSN OF VA PAC GPRT PROCEDURE
// USER=, USER CODE
// SYSPAF='$INDUV..SYSPAF', DSN PREFIX VSAM KSDS WORK FILE
// SPABJ=(TRK,(02,01),RLSE)'SPACE' OF GENERATION FILE
//*****
// *
//PRMSYS EXEC PGM=PRMSYS,PARM='&USER'
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PACRIN DD DSN=&INDSN..&ROOT.&ROOT.SY(DFSYPAF),DISP=SHR
//PACROU DD DSN=&&DFSYPAF,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
// SPACE=(TRK,(1,1),RLSE)
//SYSOUT DD SYSOUT=&OUT
// *
//DEFINE EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSIN DD DSN=&&DFSYPAF,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=&OUT
// *
//MAXKEY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPAF DD DSN=&SYSPAF..&USER,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//MAXKEY DD DSN=&INDSN..&ROOT.&ROOT.SY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(REPRO999),DISP=SHR
// *
//PTC100 EXEC PGM=PTC100,REGION=4096K
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7TS DD DSN=&INDSV..&ROOT.&ROOT.TS,DISP=SHR
//SYSPAF DD DSN=&SYSPAF..&USER,DISP=SHR
//PAC7JB DD DSN=&PAC7GB,DISP=SHR
//PAC7BJ DD DSN=&&PAC7BJ,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
// SPACE=&SPABJ
//PAC7ET DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
// *
//IEBGN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=&&PAC7BJ,DISP=(OLD,PASS)
//SYSUT2 DD SYSOUT=(&OUT,INTRDR)
//SYSIN DD DUMMY
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
// *
//PRMSYS EXEC PGM=PRMSYS,PARM='&USER',COND=EVEN
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PACRIN DD DSN=&INDSN..&ROOT.&ROOT.SY(DLSYPAF),DISP=SHR
//PACROU DD DSN=&&DLSYPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)

```

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION

1

```
//SYSOUT DD SYSOUT=&OUT
//*
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=&OUT
/*
```

	PAGE	212
VERSIONING UTILITIES		3
TEAMCONNECTION		3
TCCI: INTER-ENVIRONMENT INTEGRITY CHECK		2

3.3.2. TCCI: INTER-ENVIRONMENT INTEGRITY CHECK
3.3.2.1. TRCI: INTRODUCTION

TCCI: INTRODUCTION

The TCCI procedure checks the consistency of TeamConnection information stored in VisualAge Pacbase, as well as the presence of VA Pac elements in these environments. It produces appropriate VA Pac 'correction' transactions.

EXECUTION CONDITION

The interface must not be currently open.

VERSIONING UTILITIES
TEAMCONNECTION
TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

PAGE

213

3
3
2

3.3.2.2. TRCI: DESCRIPTION OF STEPS

EXTRACTION OF OCCURRENCES WITH TYPE CODE \$7B: PTC400

.Permanent input/output files:
-LIBRARY and SESSION file
PAC7TS: DSN=&INDSV..&ROOT.&ROOT.TS
-Error messages
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE
-Data file
PAC7AR: DSN=&INDSV..&ROOT.&FILE.AR
-Index file
PAC7AN: DSN=&INDSV..&ROOT.&FILE.AN

.Input file:
-Manager's identification
PAC7CA: DSN=&&TCCIMB

.Work file:
-PAF standard KSDS file
SYSPAF: DSN=&SYSPAF..&USER

.Output report:
-Detected errors
PAC7ET

.Output file:
-Extracted transactions
PAC7RT: DSN=&&PAC7RT

INTER-ENVIRONMENT CHECK: PTC440

.Permanent input files:
-LIBRARY and SESSION file
PAC7TS: DSN=&INDSV..&ROOT.&ROOT.TS
-Error messages
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

.Input files:
-Manager's identification
PAC7CA: DSN=&&TCCIMB
-List of generated objects found in TeamConnection
PAC7UN: DSN=&&PAC7RS
-List of generated objects stored in VA Pac
PAC7UM: DSN=&&PAC7RT

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

2

.Output report:
-Detected errors
PAC7ET

.Output file:
-'Correction'-transaction file
PAC7UR: DSN=&&PAC7UR

GENERATION OF BATCH UPDATE TRANSACTIONS: PTC220

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

.Input file:
-'Correction' transactions
PAC7UR: DSN=&&PAC7UR

.Work file:
-PAF standard KSDS file
SYSPAF: DSN=&SYSPAF..&USER

.Output report:
-Check report
PAC7ET

.Output file:
-VA-Pacbase update transactions
PAC7MV: DSN=&&PAC7MV

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

2

3.3.2.3. TCCI: EXECUTION

```

//*****
// * VisualAge Pacbase
// *
// * -- TEAMCONNECTION - INTER-ENVIRONMENT INTEGRITY CHECK --
//*****
//$RADP.TCCI PROC FILE=$FILE, VA PAC PHYSICAL DATABASE NUMBER
// ROOT=$ROOT, VA PAC SYSTEM ROOT
// INDSV='$INDSV', VSAM SYSTEM FILES INDEX
// INDSN='$INDSN', NON-VSAM SYSTEM FILES INDEX
// INDUV='$INDUV', VSAM USER FILES INDEX
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LOAD-MODULES LIBRARY
// SORTLIB='$BIBT', SORT LIBRARY
// CYL=1, SIZE OF SORT WORKS
// OUT=$OUT, PRINTOUT CLASS
// UWK=$UWK, WORK UNIT
// USER=, USER CODE
// PAC7UN=, TEAMCONNECTION PARTS FILE
// SYSPAF='$INDUV..SYSPAF', WORK KSDS VSAM PREFIX DSN
// SPART='(TRK,(02,01),RLSE)', 'SPACE' OF TEMP. FILES
// SPAMV='(TRK,(05,01),RLSE)' 'SPACE' OF UPDATE TRANS. FILE
//*****
//*
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&TCCIMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*
//PRMSYS EXEC PGM=PRMSYS,PARM='&USER'
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PACRIN DD DSN=&INDSN..&ROOT.&ROOT.SY(DFSYSYSPAF),DISP=SHR
//PACROU DD DSN=&&DFSYSYSPAF,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
// SPACE=(TRK,(1,1),RLSE)
//SYSOUT DD SYSOUT=&OUT
//*
//DEFINE EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSIN DD DSN=&&DFSYSYSPAF,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=&OUT
//*
//MAXKEY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPAF DD DSN=&SYSPAF..&USER,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//MAXKEY DD DSN=&INDSN..&ROOT.&ROOT.SY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(REPRO999),DISP=SHR
//*
//PTC400 EXEC PGM=PTC400,REGION=4096K
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7TS DD DSN=&INDSV..&ROOT.&ROOT.TS,DISP=SHR
//SYSPAF DD DSN=&SYSPAF..&USER,DISP=SHR
//PAC7CA DD DSN=&&TCCIMB,DISP=(OLD,PASS)
//PAC7RT DD DSN=&&PAC7RT,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=176,BLKSIZE=26400),
// SPACE=&SPART
//PAC7ET DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

2

```

//*
//PTC440 EXEC PGM=PTC440
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7TS DD DSN=&INDSV..&ROOT.&ROOT.TS,DISP=SHR
//PAC7CA DD DSN=&&TCCIMB,DISP=(OLD,PASS)
//PAC7UM DD DSN=&&PAC7RT,DISP=(OLD,PASS)
//PAC7UN DD DSN=&PAC7UN,DISP=SHR
//PAC7UR DD DSN=&&PAC7UR,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=189,BLKSIZE=18900),
// SPACE=&SPAMV
//PAC7ET DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PTC220 EXEC PGM=PTC220
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7UR DD DSN=&&PAC7UR,DISP=(OLD,PASS)
//SYSPAF DD DSN=&SYSPAF..&USER,DISP=SHR
//PAC7MV DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=8800),
// SPACE=&SPAMV
//PAC7ET DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
//PRMSYS EXEC PGM=PRMSYS,PARM='&USER',COND=EVEN
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//PACRIN DD DSN=&INDSN..&ROOT.&ROOT.SY(DLSYSPAF),DISP=SHR
//PACROU DD DSN=&&DLSYSPAF,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSOUT DD SYSOUT=&OUT
//*
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSIN DD DSN=&&DLSYSPAF,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=&OUT
//*
```


	PAGE	217
VERSIONING UTILITIES		3
TEAMCONNECTION		3
TCLS: LIBRARY-SESSION UPDATE		3

3.3.3. TCLS: LIBRARY-SESSION UPDATE
3.3.3.1. TCLS: INTRODUCTION

TCLS: INTRODUCTION

The TCLS procedure updates the Target Library and Session file used by the VA Pac / TeamConnection Bridge.

EXECUTION CONDITION

None.

VERSIONING UTILITIES
 TEAMCONNECTION
 TCLS: LIBRARY-SESSION UPDATE

3
 3
 3

3.3.3.2. TCLS: USER INPUT

TCLS : USER INPUT

One line for each update request.

Parameter line for target session update:

```
-----
!Pos.! Lon.! Value  ! Meaning
!-----+-----+-----+-----!
!  1 !   1 !   !   ! Transaction code
!   !   ! 'C'  ! Creation
!   !   ! 'M'  ! Modification
!   !   ! 'A'  ! Deletion
!   !   ! 'X'  ! Creation or Modification
!  6 !   2 ! 'NS' ! Line code
!  8 !   4 ! ssss ! Target-session number
! 12 !   1 !   ! Target-session status
!   !   ! 'Z'  ! Current session: '9999'
!   !   ! 'T'  ! Frozen session
! 13 !   3 ! nnn  ! Line number
! 16 !   5 ! sssst ! Beginning-session number
! 21 !   5 ! sssst ! End-session number
! 26 !  36 !   ! Comments
-----
```

The status of beginning- and end- sessions may be 'Z' or 'T'.
 Status 'T' is included in status 'Z' if the beginning- and end- sessions are one and the same.

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCLS: LIBRARY-SESSION UPDATE

3

Parameter line for update of target libraries:

```

-----
!Pos.! Len.! Value  ! Meaning                                     !
!-----+-----+-----+-----!
!  1 !   1 !         ! Transaction code                             !
!   !   ! 'C'     ! Creation                                       !
!   !   ! 'M'     ! Modification                                    !
!   !   ! 'A'     ! Deletion                                        !
!   !   ! 'X'     ! Creation or Modification                       !
!  6 !   2 ! 'NB'    ! Line code                                       !
!  8 !   3 ! bbb     ! Target-library code                            !
! 13 !   3 ! nnn     ! Line number                                     !
! 16 !   3 ! bbb     ! VA-Pacbase generation-library code           !
! 19 !  36 !         ! Comments                                       !
-----

```

PRINTED RESULT

The result printed is a report containing detected errors,
and a printout of the list of TARGET SESSIONS and LIBRARIES
defined on the site.

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCLS: LIBRARY-SESSION UPDATE

3

3.3.3.3. TCLS: DESCRIPTION OF STEPS

LIBRARY- AND SESSION- FILE UPDATE: PTC010

.Permanent input/output file:
-TARGET LIBRARY and SESSION file
PAC7TS: DSN=&INDSV..&ROOT.&ROOT.TS

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

.Input transaction file:
-User input
PAC7MV: DSN=&&TCLSMB

.Output report:
-Update report
PAC7ET

PRINTING OF LIBRARY AND SESSION FILE: PTC030

.Permanent input files:
-TARGET LIBRARIES and SESSIONS
PAC7TS: DSN=&INDSV..&ROOT.&ROOT.TS
-Error messages
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

.Output report:
-List of TARGET LIBRARY and SESSIONS
PAC7ET

VERSIONING UTILITIES
 TEAMCONNECTION
 TCLS: LIBRARY-SESSION UPDATE

3
 3
 3

3.3.3.4. TCLS: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// *
// * - TEAMCONNECTION - LIBRARY AND SESSION UPDATE -
//*****
//$RADP.TCLS PROC ROOT=$ROOT, SYSTEM ROOT
// INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LOAD-MODULES LIBRARY
// SORTLIB='$BIPT', SORT LIBRARY
// CYL=1, SORT WORKS SIZE
// OUT=$OUT, PRINTOUT CLASS
// UWK=$UWK WORK UNIT
//*****
// *
//INPUT EXEC PGM=PTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&TCLSMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
// *
//PTC010 EXEC PGM=PTC010
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,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 DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7TS DD DSN=&INDSV..&ROOT.&ROOT.TS,DISP=SHR
//PAC7MV DD DSN=&&TCLSMB,DISP=(OLD,DELETE)
//PAC7ET DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
// *
//PTC030 EXEC PGM=PTC030
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7TS DD DSN=&INDSV..&ROOT.&ROOT.TS,DISP=SHR
//PAC7ET DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
// *

```

VisualAge Pacbase - Operation Manual
BATCH PROC.: ADMINISTRATOR'S GUIDE
MANAGER'S UTILITIES

PAGE 222

4

4. MANAGER'S UTILITIES

	PAGE	223
MANAGER'S UTILITIES		4
STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES		1
STOP: INTRODUCTION		1

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.

MANAGER'S UTILITIES

4

STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES

1

STOP: USER INPUT

2

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 !gggggggg! 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.

MANAGER'S UTILITIES

4

STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES

1

STOP: DESCRIPTION OF STEPS

3

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=&INDUN..&ROOT.&FILE.PC(0)
If backup Dispatch option :
-Sequential image #2 of the database
PAC7PD: DSN=&INDUN..&ROOT.&FILE.PD(0)

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

.Output file:
-Sequential image of the database
PAC7CP: DSN=&INDUN..&ROOT.&FILE.PC(+1)
If Dispatch backup option :
-Sequential image #2 of the database
PAC7DP: DSN=&INDUN..&ROOT.&FILE.PD(+1)

.Sort files:
SORTWK01
SORTWK02
SORTWK03

.Output report:
-Execution report
PAC7EU

MANAGER'S UTILITIES

STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES

STOP: EXECUTION JCL

4

1

4

4.1.4. STOP: EXECUTION JCL

```

//*****
//* VisualAge Pacbase
//*
//*          - STORAGE OPTIMIZATION OF PACBASE DATA -
//*          (FOR MULTI VOLUMES AR FILE)
//*****
//$RADP.STOP PROC ROOT=$ROOT,          ROOT OF THE VA PAC SYSTEM
//          FILE=$FILE,                NUMBER OF PHYSICAL DATABASE
//          INDUN=$INDUN',             INDEX OF SYSTEM NON-VSAM FILES
//          STEPLIB='$MODB',           LIBRARY OF LOAD-MODULES
//*:      SYSTCAT='$SCAT',             SYSTEM VSAM CATALOG
//          OUT=$OUT,                 OUTPUT CLASS
//          SORTLIB='$BIBT',          SORT LIBRARY
//          CYL=10,                   SIZE OF WORK FILE
//          UWK=$UWK,                 WORK UNIT
//          VOLS='SER=$VOLUN',        BATABASE BACKUP VOLUME
//          UNITS=$UNITUN,            BACKUP UNIT (DISK OR CARTRIDGE)
//          VOLS2='SER=$VOLUN',       BATABASE BACKUP 2 VOLUME
//          UNITS2=$UNITUN,           BACKUP 2 UNIT (DISK OR CARTRIDGE)
//          DSCB='$DSCB',             DSCB MODEL FILE
//          SPAPC='(TRK,(100,10),RLSE)', BACKUP 'SPACE'
//          SPAPD='(TRK,(100,10),RLSE)', BACKUP 'SPACE'
//*****
//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
//PAC7EU DD SYSOUT=&OUT
//PAC7MB DD DSN=&&STOPMB,DISP=(OLD,DELETE)
//PAC7PC DD DSN=&INDUN..&ROOT.&FILE.PC(0),DISP=SHR
//PAC7PD DD DSN=&INDUN..&ROOT.&FILE.PD(0),DISP=SHR
//PAC7CP DD DSN=&INDUN..&ROOT.&FILE.PC(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,
//          VOL=&VOLS,
//          SPACE=&SPAPC,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7DP DD DSN=&INDUN..&ROOT.&FILE.PD(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS2,
//          VOL=&VOLS2,
//          SPACE=&SPAPD,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	227
MANAGER'S UTILITIES		4
SESSION MANAGEMENT		2
ESES - CSES: INTRODUCTION		1

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.

	PAGE	228
MANAGER'S UTILITIES		4
SESSION MANAGEMENT		2
ESES: EXTRACTION OF SESSION NUMBERS		2

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.

MANAGER'S UTILITIES

4

SESSION MANAGEMENT

2

ESES: EXTRACTION OF SESSION NUMBERS

2

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 !  
-----
```

MANAGER'S UTILITIES	
SESSION MANAGEMENT	
ESES: DESCRIPTION OF STEPS	

PAGE	230
	4
	2
	3

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=&INDSV..&ROOT.&ROOT.AE
-Data file
 PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
-Index file
 PAC7AN: DSN=&INDUV..&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.

MANAGER'S UTILITIES
 SESSION MANAGEMENT
 ESES: EXECUTION JCL

4
 2
 4

4.2.4. ESES: EXECUTION JCL

```
//*****
// * VisualAge Pacbase *
// * * *
// * - SESSION NUMBERS CORRESPONDENCE TABLE - *
//*****
//$RADP.ESES PROC FILE=$FILE, VA PAC DATABASE NUMBER
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDUV='$INDUV', INDEX OF VSAM USER FILES
// ESEFILE=, SESSION NUMBER FILE DSNAME
// VOLS='SER=$VOLUN', SESSION NUMBER FILE VOLUME
// UNITS=$UNITUN, SESSION NUMBER FILE UNIT
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// SPAMB='(TRK,(30,10))', SPACE OF EXTRACTED TRANSACTIONS
// UWK=$UWK, WORK UNIT
// OUT=$OUT OUTPUT CLASS
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&ESESMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PTUESS EXEC PGM=PTUESS
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7EU DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV.&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV.&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV.&ROOT.&FILE.AR,DISP=SHR
//PAC7MB DD DSN=&&ESESMB,DISP=(OLD,DELETE)
//PAC7MV DD DSN=&ESESEFILE,DISP=(,CATLG),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,BLKSIZE=5000,LRECL=5)
//SYSUDUMP DD SYSOUT=&OUT
```

	PAGE	232
MANAGER'S UTILITIES		4
SESSION MANAGEMENT		2
CSES: COMPRESSION OF SESSION NUMBERS		5

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=&INDSV..&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=&INDUN..&ROOT.&FILE.PC(0)
If Dispatch option of the backup:
PAC7PD: DSN=&INDUN..&ROOT.&FILE.PD(0)
-Output
PAC7CP: DSN=&INDUN..&ROOT.&FILE.PC(+1)
If Dispatch option of the backup:
PAC7DP: DSN=&INDUN..&ROOT.&FILE.PD(+1)

.Retrieval of the VisualAge Pacbase archived journal:
-Input
PAC7PJ: DSN=&INDUN..&ROOT.&FILE.PJ(0)
-Output
PAC7JP: DSN=&INDUN..&ROOT.&FILE.PJ(+1)

.Retrieval of the VA Pac generation-print request backup:
-Input
PAC7PG: DSN=&INDUN..&ROOT.&FILE.PG(0)
-Output
PAC7GP: DSN=&INDUN..&ROOT.&FILE.PG(+1)

.Retrieval of the PEI backup:
-Input
PAC7PP: DSN=&INDUN..&ROOT.&FILE.PP(0)

MANAGER'S UTILITIES

4

SESSION MANAGEMENT

2

CSES: DESCRIPTION OF STEPS

7

-Output
PAC7EP: DSN=&INDUN..&ROOT.&FILE.PP(+1)

If DSMS is installed:

.Retrieval of the DSMS database backup:

-Input
PACDBB: DSN=&DSMS.BB(0)
-Output
PACDJB: DSN=&DSMS.BB(+1)

.Retrieval of the DSMS archived journal:

-Input
PACDDJ: DSN=&DSMS.BJ(0)
-Output
PAC7JD: DSN=&DSMS.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

MANAGER'S UTILITIES
SESSION MANAGEMENT
CSES: EXECUTION JCL

4
2
8

4.2.8. CSES: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - COMPRESSION OF SESSION NUMBERS - *
//*****
//$RADP.CSES PROC FILE=$FILE, NUMBER OF THE VA PAC DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV=' $INDSV', INDEX OF SYSTEM VSAM FILES
// INDUN=' $INDUN', INDEX OF VA PAC BACKUP FILES
// DSMS=' $DSMS', INDEX OF DSMS BACKUP FILES
// ESEFILE=, SESSION NUMBERS FILE DSNAME
// DSNTC0=, TABLES BACKUP FILE DSNAME (0)
// DSNTC1=, TABLES BACKUP FILE DSNAME (+1)
// STEPLIB=' $MODB', LIBRARY OF LOAD-MODULES
// UWK=$UWK, WORK UNIT
// DSCB=' $DSCB', DSCB MODEL FILE
// OUT=$OUT, OUTPUT CLASS
// VOLS=' $SER=$VOLUN', BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT
// SPAPC=' (TRK,(100,10),RLSE)', PC BACKUP SPACE
// SPAPD=' (TRK,(100,10),RLSE)', PD BACKUP SPACE
// SPAPJ=' (TRK,(100,10),RLSE)', PJ BACKUP SPACE
// SPAPG=' (TRK,(15,5),RLSE)', PG BACKUP SPACE
// SPAPP=' (TRK,(15,5),RLSE)', PP BACKUP SPACE
// SPABB=' (TRK,(100,10),RLSE)', BB BACKUP SPACE
// SPABJ=' (TRK,(100,10),RLSE)', BJ BACKUP SPACE
// SPATC=' (TRK,(100,10),RLSE)', TC BACKUP SPACE
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&CSESMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PTUCSS EXEC PGM=PTUCSS
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7EU DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7MB DD DSN=&&CSESMB,DISP=(OLD,PASS)
//PAC7MV DD DSN=&ESEFILE,DISP=SHR
//PAC7PC DD DSN=&INDUN..&ROOT.&FILE.PC(0),DISP=SHR
//PAC7PD DD DSN=&INDUN..&ROOT.&FILE.PD(0),DISP=SHR
//PAC7CP DD DSN=&INDUN..&ROOT.&FILE.PC(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=&SPAPC,
// DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7DP DD DSN=&INDUN..&ROOT.&FILE.PD(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=&SPAPD,
// DCB=(&DSCB,RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7PJ DD DSN=&INDUN..&ROOT.&FILE.PJ(0),DISP=SHR
//PAC7JP DD DSN=&INDUN..&ROOT.&FILE.PJ(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=&SPAPJ,
// DCB=(&DSCB,RECFM=FB,BLKSIZE=6179,LRECL=167)
//PAC7PG DD DSN=&INDUN..&ROOT.&FILE.PG(0),DISP=SHR
//PAC7GP DD DSN=&INDUN..&ROOT.&FILE.PG(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,
// SPACE=&SPAPG,
// DCB=(&DSCB,RECFM=FB,BLKSIZE=6150,LRECL=150)
//PAC7PP DD DSN=&INDUN..&ROOT.&FILE.PP(0),DISP=SHR
//PAC7EP DD DSN=&INDUN..&ROOT.&FILE.PP(+1),
// DISP=(,CATLG,DELETE),

```

MANAGER'S UTILITIES

4

SESSION MANAGEMENT

2

CSES: EXECUTION JCL

8

```
//          UNIT=&UNITS ,VOL=&VOLS ,
//          SPACE=&SPAPP ,
//          DCB=( &DSCB,RECFM=FB,BLKSIZE=6160,LRECL=110)
//PACDBB   DD DSN=&DSMS.BB(0),DISP=SHR
//PACDJB   DD DSN=&DSMS.BB(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS ,VOL=&VOLS ,
//          SPACE=&SPABB,
//          DCB=( &DSCB,RECFM=FB,BLKSIZE=6160,LRECL=110)
//PACDDJ   DD DSN=&DSMS.BJ(0),DISP=SHR
//PACDJD   DD DSN=&DSMS.BJ(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS ,VOL=&VOLS ,
//          SPACE=&SPABJ,
//          DCB=( &DSCB,RECFM=FB,BLKSIZE=6160,LRECL=110)
//PAC7TC   DD DSN=&DSNTC0,DISP=SHR
//PAC7CT   DD DSN=&DSNTC1,
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS ,VOL=&VOLS ,
//          SPACE=&SPATC,
//          DCB=( &DSCB,RECFM=FB,BLKSIZE=10614,LRECL=1061)
//SYSUDUMP DD SYSOUT=&OUT
```

MANAGER'S UTILITIES	PAGE	238
GBIR: PARTITIONED DATABASE MANAGER		4
GBIR: INTRODUCTION		3
		1

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.

	PAGE	239
MANAGER'S UTILITIES		4
GBIR: PARTITIONED DATABASE MANAGER		3
GBIR: INTRODUCTION		1

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.

	PAGE	240
MANAGER'S UTILITIES		4
GBIR: PARTITIONED DATABASE MANAGER		3
GBIR: INTRODUCTION		1

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).

	PAGE	241
MANAGER'S UTILITIES		4
GBIR: PARTITIONED DATABASE MANAGER		3
CPSN: SUB-NETWORK COMPARISON		2

4.3.2. CPSN: SUB-NETWORK COMPARISON
4.3.2.1. CPSN: INTRODUCTION

CPSN: INTRODUCTION

The Sub-Network Comparison procedure (CPSN) compares two sub-networks extracted by the EXSN procedure, which may or may not belong to the same database, 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

Must be preceded by the execution of the EXSN procedure, which extracts the sub-networks to be compared.

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

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 being compared must have been extracted via the EXSN procedure.

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.

MANAGER'S UTILITIES
GBIR: PARTITIONED DATABASE MANAGER
CPSN: SUB-NETWORK COMPARISON

PAGE

243

4
3
2

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=&INDSV..&ROOT.&ROOT.AE

.Transaction file:

-User input
PAC7MB: DSN=&&CPSNMB

.Input files from EXSN:

-Master sub-network
PAC7MA: DSN=&INDUN..&ROOT.&FILE.&MA
-Slave sub-network
PAC7ES: DSN=&INDUN..&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.

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

CPSN: SUB-NETWORK COMPARISON

4

3

2

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=&INDSV..&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

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

CPSN: SUB-NETWORK COMPARISON

4
3
2

4.3.2.4. CPSN: EXECUTION JCL

```

//*****
// * VisualAge Pacbase
// *
// * - SUB NETWORK COMPARISON -
// * (LIBRARY COMPARISON UTILITY)
//*****
//$RADP.CPSN PROC ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSN='$INDSN', INDEX OF SYSTEM NON-VSAM FILES
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// MA=, 'MASTER' SUB-NETWORK FILE DSNAME
// SL=, 'SLAVE' SUB-NETWORK FILE DSNAME
//*: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// SORTLIB='$BIBT', SORT LIBRARY
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK, WORK UNIT
// CYL=10, SIZE OF WORKSORTS
// SPAMB='(TRK,(90,15),RLSE)' SPACE OF EXTRACTED TRANSACTIONS
//*****
//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=3440)
//PTU850 EXEC PGM=PTU850
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7ES DD DSN=&SL,DISP=OLD
//PAC7EU DD SYSOUT=&OUT,DCB=BLKSIZE=1330
//PAC7DD DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB DD DSN=&&CPSNMB,DISP=(OLD,DELETE)
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7MA DD DSN=&MA,DISP=OLD
//PAC7MK DD DSN=&&PAC7MK,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SYSUDUMP DD SYSOUT=&OUT
//SORTMK EXEC PGM=SORT,COND=(0,NE,PTU850)
//*****
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTOUT DD DSN=&&PAC7KM,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=90,BLKSIZE=6300)
//SORTIN DD DSN=&&PAC7MK,DISP=(OLD,DELETE)
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(SRTCPSN),DISP=SHR
//PTU855 EXEC PGM=PTU855,COND=(0,NE,PTU850)
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7EU DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7MB DD DSN=&&MB,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6240)
//PAC7MK DD DSN=&&PAC7KM,DISP=(OLD,DELETE)
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	246
MANAGER'S UTILITIES		4
GBIR: PARTITIONED DATABASE MANAGER		3
SASN: SUB-NETWORK BACKUP		3

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.

MANAGER'S UTILITIES

4

GBIR: PARTITIONED DATABASE MANAGER

3

SASN: SUB-NETWORK BACKUP

3

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.

MANAGER'S UTILITIES
GBIR: PARTITIONED DATABASE MANAGER
SASN: SUB-NETWORK BACKUP

PAGE

248

4
3
3

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=&INDSV..&ROOT.&ROOT.AE
-Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
-Index file
PAC7AN: DSN=&INDUV..&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

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

SASN: SUB-NETWORK BACKUP

4

3

3

.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=&INDSV..&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=&INDUN..&ROOT.&FILE.SN
- If Dispatch option:
- Database sequential image í2
- PAC7PD: DSN=&INDUN..&ROOT.&FILE.SN2

.Sort file(s):

- SORTWK01
- SORTWK02
- SORTWK03

.Output report:

- Execution report
- PAC7DS

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

SASN: SUB-NETWORK BACKUP

4

3

3

4.3.3.4. SASN: EXECUTION JCL

```

//*****
//* VisualAge Pacbase
//*
//* - SUB NETWORK BACKUP -
//* (LIBRARY COMPARISON UTILITY)
//*****
//$RADP.SASN PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE PACBASE SYSTEM
// INDSN=' $INDSN', INDEX OF SYSTEM NON VSAM FILES
// INDSV=' $INDSV', INDEX OF SYSTEM VSAM FILES
// INDUV=' $INDUV', INDEX OF USER VSAM FILES
// INDUN=' $INDUN', INDEX OF USER NON VSAM FILES
//*: VSAMCAT=' $VCAT', USER VSAM CATALOG
//*: SYSTCAT=' $SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB=' $MODB', LIBRARY OF LOAD-MODULES
// SORTLIB=' $BIBT', SORT LIBRARY
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', SUB NETWORK BACKUP VOLUME
// UNITS=$UNITUN, SUB NETWORK BACKUP UNIT
// VOLS2='SER=$VOLUN', SUB NETWORK 2ND BACKUP VOLUME
// UNITS2=$UNITUN, 2ND BACKUP UNIT (DISK OR CARTRIDGE)
// UWK=$UWK, WORK FILES UNIT
// CYL=(10,1)', TEMPORARY SPACE
// SPASN='(TRK,(100,10),RLSE)' SUB NETWORK BACKUP SPACE
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&SASNMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//VERIFY EXEC PGM=IDCAM5
//*****
//*:STEPSCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
//PTU130 EXEC PGM=PTU130,REGION=0K
//*****
//*:STEPSCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7DS DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7MB DD DSN=&&SASNMB,DISP=(OLD,DELETE)
//PAC7NA DD DSN=&&NA,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=&SPASN,
// DCB=(RECFM=FB,LRECL=55,BLKSIZE=6215)
//PAC7RA DD DSN=&&RA,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=&SPASN,
// DCB=(RECFM=FB,LRECL=149,BLKSIZE=27267)
//PAC7RP DD DSN=&&RP,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=&SPASN,
// DCB=(RECFM=FB,LRECL=149,BLKSIZE=27267)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUX DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PTU140 EXEC PGM=PTU140,COND=(0,NE,PTU130),REGION=0K
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPSCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT

```

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

SASN: SUB-NETWORK BACKUP

4

3

3

```

//PAC7DS DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV. .&ROOT.&ROOT.AE,DISP=SHR
//PAC7NA DD DSN=&&NA,DISP=(OLD,DELETE)
//PAC7RA DD DSN=&&RA,DISP=(OLD,DELETE)
//PAC7RP DD DSN=&&RP,DISP=(OLD,DELETE)
//PAC7SR DD DSN=&INDUN. .&ROOT.&FILE.SN,DISP=(,CATLG,DELETE),
//
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPASN,
// DCB=(RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7PD DD DSN=&INDUN. .&ROOT.&FILE.SN2,
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS2,
// VOL=&VOLS2,
// SPACE=&SPASN,
// DCB=(RECFM=VB,BLKSIZE=27280,LRECL=155)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SYSOUX DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	252
MANAGER'S UTILITIES		4
GBIR: PARTITIONED DATABASE MANAGER		3
EMSN: EXTRACTION FOR SUB-NETWORK MERGE		4

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.

MANAGER'S UTILITIES

4

GBIR: PARTITIONED DATABASE MANAGER

3

EMSN: EXTRACTION FOR SUB-NETWORK MERGE

4

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  ! uuuuuuu! User code !
! 11  !  8  ! pppppppp! User password !
! 19  !  3  ! bbb    ! Library code !
! 22  !  4  ! ssss   ! Session number (blank=current ses.) !
! 26  !  1  ! T      ! Session status if Test session !
-----

```

Batch procedure access authorization option: The control check is made on the first '*' line.

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.

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

EMSN: EXTRACTION FOR SUB-NETWORK MERGE

4

3

4

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. ! LINE ! INITIAL ! TARGET !
! CODE ! CODE ! LIBRARY ! LIBRARY !
-----
!      !      !      !      !
! *    ! *    ! AAA    !      ! NOT EXTRACTED
! R    ! *    ! XXX    !      ! EXTRACTED
! R    ! *    ! DDD    !      ! EXTRACTED
! R    ! *    ! EEE    !      ! EXTRACTED
! R    ! *    ! KKK    !      ! EXTRACTED
! R    ! *    ! RRR    !      ! EXTRACTED
! R    ! *    ! MMM    !      ! EXTRACTED
!      !      !      !      !
-----

```

MANAGER'S UTILITIES
GBIR: PARTITIONED DATABASE MANAGER
EMSN: EXTRACTION FOR SUB-NETWORK MERGE

PAGE

255

4
3
4

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=&INDUV..&ROOT.&FILE.AN
-Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
-Error message file
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

.Transaction file:
-User input
PAC7ME: DSN=&&EMSNMB

.Output file:
-Extracted sub-network
PAC7BB: DSN=&EMSNF

.Output reports:
-Lines required as MESN input
PAC7EE
-Extraction report
PAC7EU
-Batch-procedure authorization option
PAC7DD

.Sort file(s):
SORTWK01
SORTWK02
SORTWK03

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER
EMSN: EXTRACTION FOR SUB-NETWORK MERGE

PAGE

256

4
3
4

.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.

MANAGER'S UTILITIES

4

GBIR: PARTITIONED DATABASE MANAGER

3

EMSN: EXTRACTION FOR SUB-NETWORK MERGE

4

4.3.4.4. EMSN: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * *
// * - SUB NETWORK RETRIEVAL FOR MERGE - *
// * (LIBRARY COMPARISON UTILITY) *
//*****
//$RADP.EMSN PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDSN='$INDSN', INDEX OF SYSTEM NON VSAM FILES
// INDUV='$INDUV', INDEX OF USER VSAM FILES
// EMSNF=, EXTRACTED SUB-NETWORK FILE DSNAME
// *: VSAMCAT='$VSCAT', USER VSAM CATALOGUE
// *: SYSTCAT='$SCAT', SYSTEM VSAM CATALOGUE
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// SORTLIB='$BIBT', SORT LIBRARY
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK, WORK UNIT
// CYL=10, SIZE OF WORK OF SORT
// UNITS=$UNITUN, UNIT
// VOLS='$SER=$VOLUN', EXTRACTED SUB-NETWORK VOLUME
// SPAXF='(TRK,(90,15),RLSE)' SPACE OF EXTRACTED SUB-NETWORK FILE
//*****
// * INPUT : A USER AND LIBRARY LINE FOR EACH LIBRARY OF THE *
// * SUB-NETWORK TO BE EXTRACTED (99 LINES MAXIMUM) *
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&EMSNMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//PTU810 EXEC PGM=PTU810,REGION=0K
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
// *: STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// *: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7ME DD DSN=&&EMSNMB,DISP=(OLD,DELETE)
//PAC7DD DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7EE DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7EU DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7BB DD DSN=&EMSNF,DISP=(,CATLG),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAXF,
// DCB=(RECFM=FB,LRECL=140,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	258
MANAGER'S UTILITIES		4
GBIR: PARTITIONED DATABASE MANAGER		3
MESN: SUB-NETWORK MERGE		5

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.

MANAGER'S UTILITIES

4

GBIR: PARTITIONED DATABASE MANAGER

3

MESN: SUB-NETWORK MERGE

5

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-net- !
! ! ! ! work library code) !
-----

```

In case of error, the procedure is interrupted.

MANAGER'S UTILITIES

4

GBIR: PARTITIONED DATABASE MANAGER

3

MESN: SUB-NETWORK MERGE

5

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).

MANAGER'S UTILITIES
GBIR: PARTITIONED DATABASE MANAGER
MESN: SUB-NETWORK MERGE

PAGE

261

4
3
5

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=&INDUN..&ROOT.&FILE.PC(0)
-Extracted sub-network
PAC7BB: DSN=&EMSNF
-Error message file
PAC7AE: DSN=&INDSV..&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.

MANAGER'S UTILITIES

4

GBIR: PARTITIONED DATABASE MANAGER

3

MESN: SUB-NETWORK MERGE

5

4.3.5.4. MESN: EXECUTION JCL

```

//*****
//* VisualAge Pacbase
//*
//* - MERGE OF SUB NETWORK -
//* (LIBRARY COMPARISON UTILITY)
//*****
//$RADP.MESN PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDUN='$INDUN', INDEX OF USER NON VSAM FILES
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// EMSNF=, EXTRACTED SUB-NETWORK FILE DSNAME
// NEWPC=, MERGED FILE DSNAME
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
//*: SYSTCAT='$SCAT', VA PAC SYSTEM VSAM CATALOG
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK, WORK UNIT
// UNITS=$UNITUN, MERGED FILE UNIT
// VOLS='SER=$VOLUN', MERGED FILE VOLUME
// SPAPC='(TRK,(100,10),RLSE)' SPACE OF MERGED FILE
//*****
//* INPUT : A USER AND LIBRARY LINE FOR EACH LIBRARY OF THE
//* SUB-NETWORK TO BE MERGED (99 LINES MAXIMUM)
//* A LIST OF LINES TO BE PUT IN INPUT OF THIS PROCEDURE
//* IS PRINTED BY 'EMSN'.
//* WHEN A LIBRARY CODE IS DIFFERENT IN THE MERGED NETWORK
//* ENTER THIS LIBRARY CODE IN 'TARGET LIBRARY CODE'.
//*****
//* COL. 1 : 'R' LIBRARY TO BE MERGED
//* ' ' LIBRARY OF A HIGHER LEVEL NOT TO BE MERGED
//* COL. 2 : '*'
//* COL. 3-5 : LIBRARY CODE IN THE 'MASTER' SUB-NETWORK
//* COL. 6-8 : TARGET LIBRARY CODE IN THE 'SLAVE' SUB-NETWORK
//* (DEFAULT OPTION: 'MASTER' SUB-NETWORK LIBRARY CODE)
//*****
//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=3440)
//PTU815 EXEC PGM=PTU815
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7ME DD DSN=&&MESNMB,DISP=(OLD,DELETE)
//PAC7BB DD DSN=&EMSNF,DISP=OLD
//PAC7PC DD DSN=&INDUN..&ROOT.&FILE.PC(0),DISP=OLD
//PAC7CP DD DSN=&NEWPC,DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPC,
// DCB=(RECFM=VB,BLKSIZE=27280,LRECL=155)
//PAC7EU DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

MANAGER'S UTILITIES	PAGE	263
LOAE: AE - AP RELOADING		4
LOAE: INTRODUCTION		4
		1

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.

MANAGER'S UTILITIES
LOAE: AE - AP RELOADING
LOAE: USER INPUT

PAGE

264

4
4
2

4.4.2. LOAE: USER INPUT

LOAE: INPUT-PROCESSING-RESULTS

One compulsory line:

```
+-----+  
! Pos. ! Len. ! Value ! Meaning !  
+-----+  
! 2 ! 6 ! 'NRREST' ! Line code !  
+-----+
```


MANAGER'S UTILITIES

4

LOAE: AE - AP RELOADING

4

LOAE: DESCRIPTION OF STEPS

3

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=&INDSN..&ROOT.&ROOT.PE(0)
-Initial sequential image of error messages
  PAC7LE: DSN=&INDSN..&ROOT.&ROOT.AE0

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

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

.Sort file(s):
SORTWK01
SORTWK02
SORTWK03

.Output report:
-Reconstruction report
  PAC7IJ
```

MANAGER'S UTILITIES

LOAE: AE - AP RELOADING

LOAE: EXECUTION JCL

4

4

4

4.4.4. LOAE: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - LOADING OF USER PARAMETERS - *
//*****
//$RADP.LOAE PROC ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDSN='$INDSN', INDEX OF SYSTEM NON VSAM FILES
//*: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// SORTLIB='$BIBT', SORT LIBRARY
// OUT=$OUT, OUTPUT CLASS
// UWK=$UWK, WORK FILES UNIT
// CYL=1 SORTWORK SPACE
//*****
//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=3440)
//VERIFY EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AP DD DSN=&INDSV..&ROOT.&ROOT.AP,DISP=OLD
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=OLD
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAP),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAE),DISP=SHR
//DEFINE EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AE),
// DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(DF&ROOT.&ROOT.AP),
// DISP=SHR
//PACU80 EXEC PGM=PACU80,REGION=0K
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//PAC7IJ DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=OLD
//PAC7AP DD DSN=&INDSV..&ROOT.&ROOT.AP,DISP=OLD
//PAC7LE DD DSN=&INDSN..&ROOT.&ROOT.AE0,DISP=SHR
//PAC7MC DD DSN=&&LOAEMB,DISP=(OLD,DELETE)
//PAC7CE DD DSN=&INDSN..&ROOT.&ROOT.PE(0),DISP=OLD
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	267
MANAGER'S UTILITIES		4
VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY		5
VINS: INTRODUCTION		1

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,
- Is 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.

MANAGER'S UTILITIES

4

VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY

5

VINS: DESCRIPTION OF STEPS

3

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

ARLSR: DSN=&INDUV..&ROOT.&FILE.AR

PAC7AR: SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')

-Index file

ANLSR: DSN=&INDUV..&ROOT.&FILE.AN

PAC7AN:

SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')

-Journal file

PAC7AJ: DSN=&INDUV..&ROOT.&FILE.AJ

.Permanent input file:

-Error message file

PAC7AE: DSN=&INDSV..&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

Note: If you cannot use the Lsr batch system option in the PACINS program, delete the calls to Lsr batch (DD SUBSYS) and replace ANLSR with PAC7AN and ARLSR with PAC7AR.

MANAGER'S UTILITIES

4

VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY

5

VINS: EXECUTION JCL

4

4.5.4. VINS: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - DICTIONARY UPDATING WITH C.G.I. ENTITIES - *
//*****
//$RADP.VINS PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDSN='$INDSN', INDEX OF SYSTEM NON VSAM FILES
// INDUV='$INDUV', INDEX OF USER VSAM FILES
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
//*: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// FDIC='$INDSN..$ROOT.$ROOT.VGE', VA SMLTK DICT. DSNAME
// LSR='BLSR', LSR BATCH SYSTEM NAME
// UWK=$UWK WORK UNIT
//*****
//INPUT EXEC PGM=PTU001
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&VINSMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// 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
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&ROOT.&FILE.AJ,DISP=SHR
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..&ROOT.&ROOT.SY(VERIFAJ),DISP=SHR
//PACINS EXEC PGM=PACINS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7EE DD SYSOUT=&OUT
//PAC7IE DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//ANLSR DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AN DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7AR DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7AJ DD DSN=&INDUV..&ROOT.&FILE.AJ,DISP=SHR
//PAC7MB DD DSN=&VINSMB,DISP=(OLD,DELETE)
//PAC7MV DD DSN=&FDIC,DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT

```

MANAGER'S UTILITIES	PAGE	271
RTLO: DELETION OF INVALID UPDATE LOCKS		4
RTLO: INTRODUCTION		6
		1

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.

MANAGER'S UTILITIES

RTLO: DELETION OF INVALID UPDATE LOCKS

RTLO: DESCRIPTION OF STEPS

PAGE

272

4

6

2

4.6.2. RTLO: DESCRIPTION OF STEPS

RTLO: DESCRIPTION OF STEPS

RETRIEVAL OF INVALID LOCKS: PTULOI

.Permanent Input files:
-Error-message file
PAC7AE: DSN=&INDSV..&ROOT.&ROOT.AE

.Permanent Input/Output files:
-Data file
PAC7AR: DSN=&INDUV..&ROOT.&FILE.AR
-Index file
PAC7AN: DSN=&INDUV..&ROOT.&FILE.AN

.Output file:
-Generated deletion transactions
PAC7MB: DSN=&&RTLoupdt

.Output report:
-Lists
PAC7EU

.Internal Sort:
SORTWK01
SORTWK02
SORTWK03

MANAGER'S UTILITIES

RTLO: DELETION OF INVALID UPDATE LOCKS

RTLO: EXECUTION JCL

4

6

3

4.6.3. RTLO: EXECUTION JCL

```

//*****
//* VisualAge Pacbase *
//* *
//* - RETRIEVAL OF LOCKED ENTITIES - *
//*****
//$RADP.RTLO PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDUV='$INDUV', INDEX OF USER VSAM FILES
//*: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
//*: VSAMCAT='$VCAT', USER VSAM CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// SORTLIB='$BIBT', SORT LIBRARY
// OUT=$OUT, OUTPUT CLASS
// SPAMV='(TRK,(30,10))', SPACE OF EXTRACTED TRANSACTIONS
// CYL=5, SIZE OF WORK OF SORT
// UWK=$UWK WORK UNIT
//*****
//PTULOI EXEC PGM=PTULOI
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..&ROOT.&ROOT.AE,DISP=SHR
//PAC7AN DD DSN=&INDUV..&ROOT.&FILE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&ROOT.&FILE.AR,DISP=SHR
//PAC7EU DD SYSOUT=&OUT
//PAC7MB DD DSN=&RTLOUPDT,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL),,CONTIG)
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

MANAGER'S UTILITIES	PAGE	274
UXSR: PARTIAL SUB-NETWORK EXTRACTION		4
UXSR: INTRODUCTION		7
		1

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*APPENBBB (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.

MANAGER'S UTILITIES

UXSR: PARTIAL SUB-NETWORK EXTRACTION

UXSR: USER INPUT

4

7

2

4.7.2. UXSR: USER INPUT

UXSR: USER INPUT

One '*' line:

```

-----
!Pos.! Len.! Value  ! Meaning
!-----+-----+-----+-----!
!  2 !   1 !  '*'  ! Line code
!  3 !   8 ! uuuuuuu ! 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).

MANAGER'S UTILITIES
UXSR: PARTIAL SUB-NETWORK EXTRACTION
UXSR: DESCRIPTION OF STEPS

PAGE

277

4
7
3

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

MANAGER'S UTILITIES

UXSR: PARTIAL SUB-NETWORK EXTRACTION

UXSR: EXECUTION JCL

4

7

4

4.7.4. UXSR: EXECUTION JCL

```

//*****
// * VisualAge Pacbase *
// * * *
// * - EXTRACTION FOR SUB-NETWORK UPDATE - *
//*****
// $RADP.UXSR PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDSV=' $INDSV', INDEX OF SYSTEM VSAM FILES
// INDSN=' $INDSN', INDEX OF SYSTEM NON VSAM FILES
// INDUV=' $INDUV', INDEX OF USER VSAM FILES
// INDUN=' $INDUN', INDEX OF USER NON VSAM FILES
// * : VSAMCAT=' $VCAT', USER VSAM CATALOG
// * : SYSTCAT=' $SCAT', VA PAC SYSTEM VSAM CATALOG
// STEPLIB=' $MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// VOLS=' $SER=$VOLUN', DATABASE BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// UWK=$UWK, WORK UNIT
// SPAPC=' (TRK,(100,10),RLSE)' VA PAC DATABASE SPACE
//*****
// * *
// * INPUTS : *
// * *
// * DETAIL LINE (ONE PER LIBRARY) : *
// * COL 1 : 'C' *
// * COL 2 : '*' *
// * COL 3 TO 5 : CODE OF THE LIBRARY TO BE CREATED *
// * COL 6 TO 8 : CODE OF THE CENTRAL LIBRARY *
// * COL 9 TO 11: CODE OF THE SOURCE LIBRARY *
// * *
//*****
// INPUT EXEC PGM=PTU001
//*****
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// CARTE DD DDNAME=SYSIN
// 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=&VSAMCAT,DISP=SHR
// SYSPRINT DD SYSOUT=&OUT
// PAC7AR DD DSN=&INDUV. &ROOT. &FILE. AR,DISP=SHR
// SYSIN DD DSN=&INDSN. &ROOT. &ROOT. SY (VERIFAR),DISP=SHR
// UTIXSR EXEC PGM=UTIXSR
//*****
// * : STEPCAT DD DSN=&VSAMCAT,DISP=SHR
// * : DD DSN=&SYSTCAT,DISP=SHR
// STEPLIB DD DSN=&STEPLIB,DISP=SHR
// SYSOUT DD SYSOUT=&OUT
// PAC7AE DD DSN=&INDSV. &ROOT. &ROOT. AE,DISP=SHR
// PAC7AR DD DSN=&INDUV. &ROOT. &FILE. AR,DISP=SHR,
// AMP=(' BUFND=3')
// PAC7MB DD DSN=&UXSRMB,DISP=(OLD,DELETE)
// PAC7PC DD DSN=&INDUN. &ROOT. &FILE. UXSR,
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPC,
// DCB=(RECFM=VB,BLKSIZE=27280,LRECL=155)
// PAC7DD DD SYSOUT=&OUT
// PAC7EU DD SYSOUT=&OUT
// PAC7EV DD SYSOUT=&OUT
// SYSUDUMP DD SYSOUT=&OUT

```

VisualAge Pacbase - Operation Manual
BATCH PROC.: ADMINISTRATOR'S GUIDE
MIGRATIONS

PAGE 279

5

5. MIGRATIONS

MIGRATIONS	PAGE	280
CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS		5
CRYP: INTRODUCTION		1
		1

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).

MIGRATIONS	5
CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS	1
CRYP: USER INPUT	2

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.

MIGRATIONS

CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS
CRYP: DESCRIPTION OF STEPS

PAGE

282

5
1
3

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

MIGRATIONS

5

CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS

1

CRYP: EXECUTION JCL

4

5.1.4. CRYP: EXECUTION JCL

```

//*****
//* VisualAge Pacbase *
//* * *
//* - CODING-DECODING OF USER PASSWORDS - *
//* FROM THE USER PARAMETERS BACKUP *
//*****
//$RADP.CRYP PROC OUT=$OUT, OUTPUT CLASS
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// INFILE=, INPUT FILE DSNAME
// OUTFILE=, OUTPUT FILE DSNAME
// VOLS='SER=$VOLSN', OUTPUT FILE VOLUME
// UNITS=$UNITSN, OUTPUT FILE UNIT (DISK OR CARTRIDGE)
// UWK=$UWK, WORK UNIT
// SPACY='(TRK,(15,1),RLSE)' OUTPUT FILE SPACE
//*****
//* FORMAT OF TRANSACTIONS AT INPUT : *
//* .. A * USER CODE AND PASSWORD LINE *
//* (AUTHORIZATION LEVEL 4 ON PARAMETERS) *
//* .. A COMMAND LINE *
//* COL 3-8 : 'CODE ' *
//* : 'DECODE' *
//*****
//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,
// VOL=&VOLS,
// SPACE=&SPACY,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSUDUMP DD SYSOUT=&OUT

```

MIGRATIONS	PAGE	284
LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE		5
LVBL: INTRODUCTION		2
		1

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

	PAGE	285
MIGRATIONS		5
LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE		2
LVBL: DESCRIPTION OF STEPS		2

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=&INDUN..&ROOT.&FILE.PC(0)

.Output file:

-Database backup
PAC7PC: DSN=&INDUN..&ROOT.&FILE.PC(+1)

MIGRATIONS

LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE

LVBL: EXECUTION JCL

5

2

3

5.2.3. LVBL: EXECUTION JCL

```

//*****
//* VisualAge Pacbase *
//* *
//* - CHANGE LOW VALUE CHARACTERS INTO BLANK CHARACTERS - *
//* OPTION : SUBMIT PROCEDURE WITH PARM=(DATA) TO PROCESS DATA ONLY *
//*****
//$RADP.LVBL PROC FILE=$FILE, NUMBER OF PHYSICAL DATABASE
// ROOT=$ROOT, ROOT OF THE VA PAC SYSTEM
// INDUN='$INDUN', INDEX OF USER NON VSAM FILES
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// SPAPC='(TRK,(100,10),RLSE)', SPACE OF PC BACKUP
// DSCB='$DSCB' DSCB MODEL FILE
//*****
//PTULVB EXEC PGM=PTULVB,PARM=' '
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7MC DD DSN=&INDUN..&ROOT.&FILE.PC(0),DISP=SHR
//PAC7PC DD DSN=&INDUN..&ROOT.&FILE.PC(+1),
// DISP=(,CATLG,DELETE),
// UNIT=&UNITS,
// VOL=&VOLS,
// SPACE=&SPAPC,
// DCB=(&DSCB,RECFM=VB,LRECL=155,BLKSIZE=27280)
//SYSUDUMP DD SYSOUT=&OUT

```

	PAGE	287
MIGRATIONS		
SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION		5
SMTD: INTRODUCTION		3
		1

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.

MIGRATIONS	PAGE	288
SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION		5
SMTD: DESCRIPTION OF STEPS		3
		2

5.3.2. SMTD: DESCRIPTION OF STEPS

SMTD: DESCRIPTION OF STEPS

TD BACKUP: PTASVD

.Permanent input file:
-Table-description file
PAC7TD: DSN=&TABTDF

.Output file:
-Table-description backup for migration
PAC7TC: DSN=&TDSAV

MIGRATIONS

5

SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION

3

SMTD: EXECUTION JCL

3

5.3.3. SMTD: EXECUTION JCL

```

//*****
//* VisualAge Pacbase *
//* *
//* - BACKUP OF TD FILE FOR MIGRATION - *
//*****
//$RADP.SMTD PROC TABTDF=$TABTDF, DSN OF TABLES DESCRIPTION FILE
// TDSAV=, DSN OF TABLES DESCRIPTION BACKUP FILE
//*: VSAMCAT='$VCAT', VSAM USER CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT (DISK OR CARTRIDGE)
// SPATD='(TRK,(150,10))' BACKUP SPACE
//*****
//PTASVD EXEC PGM=PTASVD
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPDAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7TD DD DSN=&TABTDF,
// DISP=SHR
//PAC7TC DD DSN=&TDSAV,
// UNIT=&UNITS,
// VOL=&VOLS,
// DISP=(,CATLG,DELETE),SPACE=&SPATD,
// DCB=(RECFM=VB,LRECL=1061,BLKSIZE=10614)
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

MIGRATIONS	PAGE	290
RMTD: RESTORATION OF TABLE DESCRIPTIONS		5
RMTD: INTRODUCTION		4
		1

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.

MIGRATIONS	PAGE	291
RMTD: RESTORATION OF TABLE DESCRIPTIONS		5
RMTD: DESCRIPTION OF STEPS		4
		2

5.4.2. RMTD: DESCRIPTION OF STEPS

RMTD: DESCRIPTION OF STEPS

TD FILE RESTORATION: PTARSD

.Input backup file:
-Table-Description sequential file
PAC7TC: DSN=&TDSAV

.Output file:
-Table-Description file
PAC7TD: DSN=&TABTDF

MIGRATIONS

5

RMTD: RESTORATION OF TABLE DESCRIPTIONS

4

RMTD: EXECUTION JCL

3

5.4.3. RMTD: EXECUTION JCL

```

//*****
//* VisualAge Pacbase *
//* * *
//* - RESTORE TD FILE AFTER MIGRATION - *
//*****
//$RADP.RMTD PROC TABTDF=$TABTDF, DSN OF TABLES DESCRIPTION FILE
//      ROOT='$ROOT', VA PAC SYSTEM ROOT
//      TDSAV=, DSN OF TABLES DESCRIPTION BACKUP FILE
//      INDSN='$INDSN', NON VSAM SYSTEM FILES INDEX
//*:      VSAMCAT='$VCAT', VSAM USER CATALOG
//      OUT=$OUT, OUTPUT CLASS
//*****
//DELDEF EXEC PGM=IDCAMS
//*****
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..&ROOT.&ROOT.SY(DFTABTDF),DISP=SHR
//PTARSD EXEC PGM=PTARSD
//*****
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//PAC7TC DD DSN=&TDSAV,DISP=OLD
//PAC7TD DD DSN=&TABTDF,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT

```

MIGRATIONS	
RPTD: TABLE DESCRIPTIONS RETRIEVAL	
RPTD: INTRODUCTION	

PAGE	293
	5
	5
	1

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 handlings. The pivot year for century change is parameterized.

EXECUTION CONDITION

None.

PRINTOUT

The RPTD procedure prints a report on the retrieval.

MIGRATIONS

RPTD: TABLE DESCRIPTIONS RETRIEVAL

RPTD: USER INPUT

5

5

2

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 ! Pivot Year !
!   !   ! other   !         !
!   !   ! than '00'!         !
+-----+-----+-----+-----+

```

MIGRATIONS	
RPTD: TABLE DESCRIPTIONS RETRIEVAL	
RPTD: DESCRIPTION OF STEPS	

PAGE	295
	5
	5
	3

5.5.3. RPTD: DESCRIPTION OF STEPS

RPTD : DESCRIPTION DES ETAPES

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

MIGRATIONS

RPTD: TABLE DESCRIPTIONS RETRIEVAL

5

RPTD: EXECUTION JCL

5

4

5.5.4. RPTD: EXECUTION JCL

```

//*****
//* VisualAge Pacbase *
//* * *
//* - RETRIEVAL OF TD FILE FOR MIGRATION - *
//*****
//$RADP.RPTD PROC TDSAV=, DSN OF NEW TABLES DESCRIPTION BACKUP
// OLDTDSAV=, DSN OF OLD TABLES DESCRIPTION BACKUP
//*: VSAMCAT='$VCAT', VSAM USER CATALOG
// STEPLIB='$MODB', LIBRARY OF LOAD-MODULES
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', BACKUP VOLUME
// UNITS=$UNITUN, 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
//*:STEPCHAT 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

```