



VisualAge Pacbase 2.5

**VA PAC 2.5 : IBM DOS/VSE
OPERATIONS MANUAL VOLUME II : ADMINISTRATOR'S GUIDE**

DEPDO002251A

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 (June 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 |
| 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..... | 23 |
| 2.1.4. MLIB: EXECUTION JCL..... | 25 |
| 2.2. SAVE: DATABASE BACKUP | 27 |
| 2.2.1. SAVE: INTRODUCTION..... | 27 |
| 2.2.2. SAVE: PROCESSING - RESULTS | 29 |
| 2.2.3. SAVE: DESCRIPTION OF STEPS..... | 31 |
| 2.2.4. SAVE: EXECUTION JCL..... | 32 |
| 2.3. SASY: DATABASE SYSTEM BACKUP COMPLEMENT..... | 34 |
| 2.3.1. SASY: INTRODUCTION | 34 |
| 2.3.2. SASY: DESCRIPTION OF STEPS..... | 36 |
| 2.3.3. SASY: EXECUTION JCL..... | 37 |
| 2.4. REST: DATABASE RESTORATION..... | 38 |
| 2.4.1. REST: INTRODUCTION..... | 38 |
| 2.4.2. REST: USER INPUT..... | 39 |
| 2.4.3. REST: DESCRIPTION OF STEPS | 43 |
| 2.4.4. REST: EXECUTION JCL..... | 46 |
| 2.5. RESY: DATABASE SYSTEM RESTORATION COMPLEMENT | 49 |
| 2.5.1. RESY: INTRODUCTION..... | 49 |
| 2.5.2. RESY: USER INPUT - RESULTS..... | 51 |
| 2.5.3. RESY: DESCRIPTION OF STEPS | 54 |
| 2.5.4. RESY: EXECUTION JCL..... | 57 |
| 2.6. ARCH: JOURNAL ARCHIVAL | 60 |
| 2.6.1. ARCH: INTRODUCTION..... | 60 |
| 2.6.2. ARCH: INPUT - RECOMMENDATIONS - RESULTS..... | 61 |
| 2.6.3. ARCH: DESCRIPTION OF STEPS..... | 64 |
| 2.6.4. ARCH: EXECUTION JCL..... | 66 |
| 2.7. REOR: DATABASE REORGANIZATION | 68 |
| 2.7.1. REOR: INTRODUCTION..... | 68 |
| 2.7.2. REOR: INPUT - RECOMMENDATIONS | 70 |
| 2.7.3. REOR: DESCRIPTION OF STEPS | 73 |
| 2.7.4. REOR: EXECUTION JCL..... | 76 |
| 2.8. SVAG: GENERATION-PRINT REQUEST BACKUP..... | 80 |
| 2.8.1. SVAG: INTRODUCTION..... | 80 |
| 2.8.2. SVAG: DESCRIPTION OF STEPS..... | 81 |
| 2.8.3. SVAG: EXECUTION JCL..... | 82 |
| 2.9. REAG: GENERATION-PRINT REQUEST RESTORATION | 83 |
| 2.9.1. REAG: INTRODUCTION..... | 83 |
| 2.9.2. REAG: USER INPUT | 84 |
| 2.9.3. REAG: DESCRIPTION OF STEPS | 85 |
| 2.9.4. REAG: EXECUTION JCL..... | 86 |
| 2.10. PARM: UPDATE OF USER PARAMETERS..... | 87 |
| 2.10.1. PARM: INTRODUCTION | 87 |
| 2.10.2. PARM: INPUT - RECOMMENDATIONS | 89 |
| 2.10.3. PARM: USER-CODE DEFINITION | 96 |
| 2.10.4. PARM: USER-CODE GLOBAL AUTHORIZATIONS..... | 99 |
| 2.10.5. PARM: USER-CODE SPECIFIC AUTHORIZATIONS..... | 101 |
| 2.10.6. PARM: TEXT TYPES..... | 104 |
| 2.10.7. PARM: MODIFICATION OF STANDARD ERROR MESSAGES..... | 106 |
| 2.10.8. PARM: GENERATED-STREAM CONTROL CARDS | 109 |

| | |
|--|------------|
| 2.10.9. PARM: DESCRIPTION OF STEPS..... | 116 |
| 2.10.10. PARM: EXECUTION JCL..... | 118 |
| 3. VERSIONING UTILITIES | 120 |
| 3.1. PEI: PRODUCTION ENVIRONMENT INTERFACE..... | 121 |
| 3.1.1. PEI: OVERVIEW..... | 121 |
| 3.1.2. INPE: FILE INITIALIZATION | 123 |
| 3.1.2.1. INPE: INTRODUCTION..... | 123 |
| 3.1.2.2. INPE: DESCRIPTION OF STEPS..... | 124 |
| 3.1.2.3. INPE: EXECUTION JCL..... | 125 |
| 3.1.3. SVPE: FILE BACKUP..... | 127 |
| 3.1.3.1. SVPE: INTRODUCTION | 127 |
| 3.1.3.2. SVPE: DESCRIPTION OF STEPS..... | 128 |
| 3.1.3.3. SVPE: EXECUTION JCL..... | 129 |
| 3.1.4. RSPE: FILE RESTORATION | 131 |
| 3.1.4.1. RSPE: INTRODUCTION | 131 |
| 3.1.4.2. RSPE: DESCRIPTION OF STEPS..... | 132 |
| 3.1.4.3. RSPE: EXECUTION JCL..... | 133 |
| 3.1.5. PRPE: PRODUCTION ENVIRONMENT PRINTOUTS..... | 134 |
| 3.1.5.1. PRPE: INTRODUCTION | 134 |
| 3.1.5.2. PRPE: USER INPUT | 135 |
| 3.1.5.3. PRPE: DESCRIPTION OF STEPS..... | 136 |
| 3.1.5.4. PRPE: EXECUTION JCL..... | 137 |
| 3.1.6. GRPE: TRANSACTION-GENERATION FOR REORGANIZATION | 138 |
| 3.1.6.1. GRPE: INTRODUCTION..... | 138 |
| 3.1.6.2. GRPE: DESCRIPTION OF STEPS | 139 |
| 3.1.6.3. GRPE: EXECUTION JCL | 140 |
| 3.1.7. HIPE: AUTOMATIC SESSION FREEZE..... | 141 |
| 3.1.7.1. HIPE: INTRODUCTION..... | 141 |
| 3.1.7.2. HIPE: USER INPUT | 142 |
| 3.1.7.3. HIPE: DESCRIPTION OF STEPS..... | 143 |
| 3.1.7.4. HIPE: EXECUTION JCL..... | 144 |
| 3.1.8. SIPE: PRODUCTION TURNOVER SIMULATION | 146 |
| 3.1.8.1. SIPE: INTRODUCTION..... | 146 |
| 3.1.8.2. SIPE: USER INPUT..... | 147 |
| 3.1.8.3. SIPE: DESCRIPTION OF STEPS | 148 |
| 3.1.8.4. SIPE: EXECUTION JCL | 149 |
| 3.2. PAC/TRANSFER..... | 150 |
| 3.2.1. TRUP: TRANSFER-PARAMETER UPDATE..... | 152 |
| 3.2.1.1. TRUP: INTRODUCTION..... | 152 |
| 3.2.1.2. TRUP: USER INPUT..... | 155 |
| 3.2.1.3. TRUP: DESCRIPTION OF STEPS | 159 |
| 3.2.1.4. TRUP: EXECUTION JCL | 161 |
| 3.2.2. TRJC: COMPRESSION OF ARCHIVED JOURNAL..... | 163 |
| 3.2.2.1. TRJC: INTRODUCTION | 163 |
| 3.2.2.2. TRJC: USER INPUT..... | 164 |
| 3.2.2.3. TRJC: DESCRIPTION OF STEPS | 165 |
| 3.2.2.4. TRJC: EXECUTION JCL | 166 |
| 3.2.3. TRPF: TRANSFER-FILE CREATION..... | 168 |
| 3.2.3.1. TRPF: INTRODUCTION | 168 |
| 3.2.3.2. TRPF: USER INPUT | 169 |
| 3.2.3.3. TRPF: DESCRIPTION OF STEPS..... | 170 |
| 3.2.3.4. TRPF: EXECUTION JCL..... | 171 |
| 3.2.4. TRDU: DSMS-ENVIRONMENT PREPARATION..... | 172 |
| 3.2.4.1. TRDU: INTRODUCTION..... | 172 |
| 3.2.4.2. TRDU: USER INPUT..... | 174 |
| 3.2.4.3. TRDU: DESCRIPTION OF STEPS..... | 175 |
| 3.2.4.4. TRDU: EXECUTION JCL..... | 177 |
| 3.2.5. UPDATE OF DSMS FUNCTION BEFORE VA PAC UPDATE | 180 |
| 3.2.6. TRRP: GENERATION OF TRANSFER TRANSACTIONS..... | 181 |
| 3.2.6.1. TRRP: INTRODUCTION..... | 181 |
| 3.2.6.2. TRRP: USER INPUT..... | 184 |
| 3.2.6.3. TRRP: DESCRIPTION OF STEPS | 185 |
| 3.2.6.4. TRRP: EXECUTION JCL..... | 187 |
| 3.2.7. UPDATE OF THE VISUALAGE PACBASE DATABASE..... | 189 |

| | |
|---|------------|
| 3.2.8. REINITIALIZATION OF THE DSMS ENVIRONMENT..... | 190 |
| 3.3. TEAMCONNECTION | 191 |
| 3.3.1. TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION | 192 |
| 3.3.1.1. TCGP: INTRODUCTION..... | 192 |
| 3.3.1.2. TCGP: DESCRIPTION OF STEPS | 193 |
| 3.3.1.3. TCGP: EXECUTION JCL | 194 |
| 3.3.2. TCCI: INTER-ENVIRONMENT INTEGRITY CHECK | 195 |
| 3.3.2.1. TRCI: INTRODUCTION..... | 195 |
| 3.3.2.2. TRCI: DESCRIPTION OF STEPS | 196 |
| 3.3.2.3. TCCI: EXECUTION | 198 |
| 3.3.3. TCLS: LIBRARY-SESSION UPDATE..... | 200 |
| 3.3.3.1. TCLS: INTRODUCTION | 200 |
| 3.3.3.2. TCLS: USER INPUT | 201 |
| 3.3.3.3. TCLS: DESCRIPTION OF STEPS..... | 203 |
| 3.3.3.4. TCLS: EXECUTION JCL..... | 204 |
| 4. MANAGER'S UTILITIES | 205 |
| 4.1. STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES | 206 |
| 4.1.1. STOP: INTRODUCTION | 206 |
| 4.1.2. STOP: USER INPUT | 207 |
| 4.1.3. STOP: DESCRIPTION OF STEPS..... | 208 |
| 4.1.4. STOP: EXECUTION JCL..... | 209 |
| 4.2. SESSION MANAGEMENT | 210 |
| 4.2.1. ESES - CSES: INTRODUCTION..... | 210 |
| 4.2.2. ESES: EXTRACTION OF SESSION NUMBERS..... | 211 |
| 4.2.3. ESES: DESCRIPTION OF STEPS | 213 |
| 4.2.4. ESES: EXECUTION JCL..... | 214 |
| 4.2.5. CSES: COMPRESSION OF SESSION NUMBERS | 215 |
| 4.2.6. CSES: USER INPUT..... | 216 |
| 4.2.7. CSES: DESCRIPTION OF STEPS | 217 |
| 4.2.8. CSES: EXECUTION JCL..... | 219 |
| 4.3. GBIR: PARTITIONED DATABASE MANAGER | 221 |
| 4.3.1. GBIR: INTRODUCTION..... | 221 |
| 4.3.2. CPSN: SUB-NETWORK COMPARISON..... | 224 |
| 4.3.2.1. CPSN: INTRODUCTION..... | 224 |
| 4.3.2.2. CPSN: NOTES ON THE RESULTS | 225 |
| 4.3.2.3. CPSN: DESCRIPTION OF STEPS | 226 |
| 4.3.2.4. CPSN: EXECUTION JCL | 228 |
| 4.3.3. SASN: SUB-NETWORK BACKUP | 230 |
| 4.3.3.1. SASN: INTRODUCTION..... | 230 |
| 4.3.3.2. SASN: USER INPUT..... | 231 |
| 4.3.3.3. SASN: DESCRIPTION OF STEPS | 232 |
| 4.3.3.4. SASN: EXECUTION JCL | 234 |
| 4.3.4. EMSN: EXTRACTION FOR SUB-NETWORK MERGE..... | 236 |
| 4.3.4.1. EMSN: INTRODUCTION..... | 236 |
| 4.3.4.2. EMSN: USER INPUT..... | 237 |
| 4.3.4.3. EMSN: DESCRIPTION OF STEPS | 239 |
| 4.3.4.4. EMSN: EXECUTION JCL | 240 |
| 4.3.5. MESN: SUB-NETWORK MERGE..... | 241 |
| 4.3.5.1. MESN: INTRODUCTION..... | 241 |
| 4.3.5.2. MESN: USER INPUT..... | 242 |
| 4.3.5.3. MESN: DESCRIPTION OF STEPS | 244 |
| 4.3.5.4. MESN: EXECUTION JCL | 245 |
| 4.4. LOAE: AE - AP RELOADING..... | 246 |
| 4.4.1. LOAE: INTRODUCTION..... | 246 |
| 4.4.2. LOAE: USER INPUT..... | 247 |
| 4.4.3. LOAE: DESCRIPTION OF STEPS | 248 |
| 4.4.4. LOAE: EXECUTION JCL | 249 |
| 4.5. VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY..... | 250 |
| 4.5.1. VINS: INTRODUCTION..... | 250 |
| 4.5.2. VINS: USER INPUT | 251 |
| 4.5.3. VINS: DESCRIPTION OF STEPS..... | 252 |
| 4.5.4. VINS: EXECUTION JCL..... | 253 |

| | |
|---|------------|
| 4.6. RTLO: DELETION OF INVALID UPDATE LOCKS..... | 254 |
| 4.6.1. RTLO: INTRODUCTION..... | 254 |
| 4.6.2. RTLO: DESCRIPTION OF STEPS..... | 255 |
| 4.6.3. RTLO: EXECUTION JCL..... | 256 |
| 4.7. UXSR: PARTIAL SUB-NETWORK EXTRACTION | 257 |
| 4.7.1. UXSR: INTRODUCTION..... | 257 |
| 4.7.2. UXSR: USER INPUT..... | 259 |
| 4.7.3. UXSR: DESCRIPTION OF STEPS..... | 260 |
| 4.7.4. UXSR: EXECUTION JCL..... | 261 |
| 5. MIGRATIONS..... | 262 |
| 5.1. CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS | 263 |
| 5.1.1. CRYP: INTRODUCTION..... | 263 |
| 5.1.2. CRYP: USER INPUT..... | 264 |
| 5.1.3. CRYP: DESCRIPTION OF STEPS..... | 265 |
| 5.1.4. CRYP: EXECUTION JCL..... | 266 |
| 5.2. LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE..... | 268 |
| 5.2.1. LVBL: INTRODUCTION..... | 268 |
| 5.2.2. LVBL: DESCRIPTION OF STEPS..... | 269 |
| 5.2.3. LVBL: EXECUTION JCL..... | 270 |
| 5.3. SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION..... | 271 |
| 5.3.1. SMTD: INTRODUCTION..... | 271 |
| 5.3.2. SMTD: DESCRIPTION OF STEPS..... | 272 |
| 5.3.3. SMTD: EXECUTION JCL..... | 273 |
| 5.4. RMTD: RESTORATION OF TABLE DESCRIPTIONS | 274 |
| 5.4.1. RMTD: INTRODUCTION..... | 274 |
| 5.4.2. RMTD: DESCRIPTION OF STEPS | 275 |
| 5.4.3. RMTD: EXECUTION JCL..... | 276 |
| 5.5. RPTD: TABLE DESCRIPTIONS RETRIEVAL | 277 |
| 5.5.1. RPTD: INTRODUCTION..... | 277 |
| 5.5.2. RPTD: USER INPUT..... | 278 |
| 5.5.3. RPTD: DESCRIPTION OF STEPS | 279 |
| 5.5.4. RPTD: EXECUTION JCL | 280 |

VisualAge Pacbase - Operations 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 | ! |

```

-----
! 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 genera-!
!      !     !             ! ting 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.    !
!      !     !             ! The new entities created from the !
!      !     !             ! extracted entities are not locked !
!      !     !             ! after UPDT                    !
!      !     !             ! The code of the user who locked  !
!      !     !             ! the entities is kept          !
!      !     !             !                               !
! 50  !  1  !             ! TRANSFER OF THE PASSWORD on the !
!      !     !             ! extraction prodedures, in the '*-!
!      !     !             ! line at the top of the generated !
!      !     !             ! output transactions:          !
!      !     ! 'Blank'    ! Password is not transferred in the!
!      !     !             ! output file.                  !
!      !     !             ! Password is transferred.       !
!      !     !             ! NOTE: For EXTR, the '*' line is  !
!      !     !             ! transferred in the output file on-!
!      !     !             ! ly if you input 'C' in position 1.!
-----

```

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

1.2. ACCESS RIGHTS: BATCH-PROCEDURE AUTHOR. OPTION

'BATCH-PROCEDURE ACCESS AUTHORIZATION' OPTION

PRINCIPLE OF THE OPTION

This option is used to grant each user the access.

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

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

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

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

CONSEQUENCE

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

OPTION ACTIVATION

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

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

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

BATCH PROCEDURE ACCESS AUTHORIZATION TABLE

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

| ! PROCEDURE ! | ! GLOBAL ! | ! 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.

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 a USER ABEND, accompanied by a message on the SYSLST 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 following table contains the most common values for the status (STATUS) and type of operation (OP).

| ! 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 UPD | ! |
| ! 30 | ! SYSTEM ERROR | ! ! OP | ! OPEN | ! |
| ! 30 | ! SYSTEM ERROR | ! ! CL | ! CLOSE | ! |
| ! 34 | ! BOUNDARY VIOLATION(SEQUENTIAL) | ! ! D | ! DELETE | ! |
| ! 92 | ! LOGIC ERROR (FOR EXAMPLE, | ! ! R | ! READ | ! |
| ! | ! OPENING AN ALREADY OPEN FILE) | ! ! P | ! START | ! |
| ! 93 | ! FILE STILL OPEN UNDER CICS | ! ! RN | ! READ NEXT | ! |
| ! 95 | ! INVALID OR INCOMPLETE FILE | ! ! | ! | ! |
| ! | ! INFORMATION | ! ! | ! | ! |

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

- The report of the execution with // OPTION LOG,PARTDUMP
- The list of installed programs and files (Job 9INS)
- etc.

| | | |
|---------------------------------------|------|----|
| VisualAge Pacbase - Operations Manual | PAGE | 18 |
| BATCH PROC.: ADMINISTRATOR'S GUIDE | | |
| DATABASE MANAGEMENT UTILITIES | | 2 |

2. DATABASE MANAGEMENT UTILITIES

| | | |
|-------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 19 |
| MLIB: DATABASE MANAGEMENT | | 2 |
| MLIB: INTRODUCTION | | 1 |
| | | 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.

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.

| | | |
|------------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 21 |
| MLIB: DATABASE MANAGEMENT | | 2 |
| MLIB: INPUT - PROCESSING - RESULTS | | 1 |
| | | 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 | 22 |
| 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

DATABASE VALIDATION: PTU100

This program is always executed.

.Permanent input files:

-Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Index file
PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
-Printing-generation request file
(in input-output if no simulation)
PAC7AG : Physical name = \$PXU..\$PRO.\$PFI.AG
-Error message file
(in input-output)
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input transaction file:

-Update transactions
PAC7MB : Physical name = \$XW..WMLIB.INPUT

.Output files:

-Sequential image of data
PAC7RP : Physical name = \$XW..WMLIB.RP
(must have capacity for all data)
-Sequential image of indexes
PAC7NA : Physical name = \$XW..WMLIB.NA
(must have capacity for all indexes)
-Sequential image of unsorted indexes
PAC7NB : Physical name = \$XW..WMLIB.NB
-Temporary storage
PAC7RQ : Physical name = \$XW..WMLIB.RQ
(1 record)

.Output reports:

-List of user transactions
SYS003
-Report on database before and after
SYS007
-Batch-procedure authorization option
SYS005

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

. 0 : OK without simulation
. 4 : OK with simulation
. 8 : No authorization for batch procedure
error about input movements
. 12 : Network access error.In this case there are two ste
an ABEND and a DUMP.

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

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

PAGE

24

2
1
3

SEQUENTIAL-IMAGE FORMATTING: PTU120

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

.Permanent input files:

-Data file
(in input-output to update session number)
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

.Temporary files:

-The 4 output files from the preceding step.

.Output file:

-Sequential image of the database
PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPC
Tape is SYS014

If Dispatch backup option:

-Database sequential image #2
PAC7PD : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPD
Tape is SYS012

.Output reports:

-None.

DATABASE MANAGEMENT UTILITIES
 MLIB: DATABASE MANAGEMENT
 MLIB: EXECUTION JCL

2
 1
 4

2.1.4. MLIB: EXECUTION JCL

```

===MOD MLIB
* $X JOB JNM=$PRFJ.MLIB,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST3$PWL
/* *****
// JOB $PRFJ.MLIB          ***** $PROD $PREL *****
*
*          DATABASE MANAGEMENT
/* *****
/* INPUT :
/*
/* FIRST LINE (REQUIRED) :
/* COL 2      : 'G'
/* COL 3      : 'I' DATABASE INITIALIZATION
/* COL 4      : 'S' MODIFICATION SIMULATION
/*
/*
/*
/* DETAIL LINE (ONE LINE PER MODIFICATION) :
/* COL 1      : TRANSACTION CODE (C=CREA, M=MODIF, D=DELETE)
/* COL 2      : '*'
/* COL 3 A 5  : CODE OF THE LIBRARY TO BE CREATED OR
/*              CODE OF THE LIBRARY TO DELETE OR
/*              CODE OF LIB. WHOSE CENTRAL IS TO BE MODIFIED
/* COL 6 A 8  : CODE OF THE CENTRAL LIBRARY
/*
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
// ASSGN SYS003,$LST3 3TRD PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,' $VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE NA --- */
DEL          ($XW..W$MODUL..NA) CL
DEF CL       (NAME ($XW..W$MODUL..NA) -
              MODEL($MODEL) -
              REC ($RPCP $RPCS) -
              RECSZ(54 54) -
              RECFM(FB(54)) -
              ) DATA (NAME ($XW..W$MODUL..NA.D) )
/* --- DELETE-DEFINE TEMPORARY FILE NB --- */
DEL          ($XW..W$MODUL..NB) CL
DEF CL       (NAME ($XW..W$MODUL..NB) -
              MODEL($MODEL) -
              REC ($RPCP $RPCS) -
              RECSZ(54 54) -
              RECFM(FB(54)) -
              ) DATA (NAME ($XW..W$MODUL..NB.D) )
* $X SLI MEM=DF$PRO.$PFI.PC.SSLIT
* $X SLI MEM=DF$PRO.$PFI.PD.SSLIT
/* --- DELETE-DEFINE TEMPORARY FILE RP --- */
DEL          ($XW..W$MODUL..RP) CL
DEF CL       (NAME ($XW..W$MODUL..RP) -
              MODEL($MODEL) -
              REC ($RPCP $RPCS) -
              RECSZ(149 149) -
              RECFM(FB(149)) -
              ) DATA (NAME ($XW..W$MODUL..RP.D) )

```

DATABASE MANAGEMENT UTILITIES

MLIB: DATABASE MANAGEMENT

MLIB: EXECUTION JCL

2

1

4

```

/* --- DELETE-DEFINE TEMPORARY FILE      RQ --- */
DEL          ($XW..W$MODUL..RQ) CL
DEF CL      (NAME ($XW..W$MODUL..RQ) -
            MODEL($MODELFB) -
            REC (1 1) -
            RECSZ(149 149) -
            RECFM(FB(149)) -
            ) DATA (NAME ($XW..W$MODUL..RQ.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
G S
/*
/. JCLST03
* --- STEP 03 --- PTU100
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AG,'$PXU..$PRO.$PFI.AG',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7NA,'$XW..W$MODUL..NA',,VSAM
// DLBL PAC7NB,'$XW..W$MODUL..NB',,VSAM
// DLBL PAC7RP,'$XW..W$MODUL..RP',,VSAM
// DLBL PAC7RQ,'$XW..W$MODUL..RQ',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU100,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTU120
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AN,'$XW..W$MODUL..NA',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7NB,'$XW..W$MODUL..NB',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7PC ON TAPE : SYS014 ASSIGNED
// DLBL PAC7PD,'$PXU..SEQ.$PRO$PFI..NEWPD',,VSAM
/* PAC7PD ON TAPE : SYS012 ASSIGNED
// DLBL PAC7PQ,'$XW..W$MODUL..RQ',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PR,'$XW..W$MODUL..RP',,VSAM,DISP=(,DELETE,KEEP)
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTU120,SIZE=(AUTO,50K)
/. JCLST05
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PD.$SLIT
/*
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
DEL          ($XW..W$MODUL..NA) CL
DEL          ($XW..W$MODUL..NB) CL
DEL          ($XW..W$MODUL..RP) CL
DEL          ($XW..W$MODUL..RQ) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|-------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 27 |
| 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 | 28 |
| 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.:

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

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
SAVE: DATABASE BACKUP
SAVE: PROCESSING - RESULTS

PAGE

30

2
2
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

31

2
2
3

2.2.3. SAVE: DESCRIPTION OF STEPS

SAVE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

DATABASE CONSISTENCY CHECK: PTUBAS

.Permanent input files:
-Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Error message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report
-Validity report (Length=079)
 SYS005

. Return code :
 . 0 : OK
 . 12 : System error

BACKUP OF THE DATABASE: PTU500

.Permanent input then input-output file:
-Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

.Permanent input files:
-Error message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
-Index File
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN

.Input transaction file:
-User transaction
 PAC7MB : Physical name = \$XW..WSAVE.INPUT

.Output file:
-Sequential image of the database
 PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPC
 Tape is SYS015
If backup Dispatch option:
-Sequential image 2 of the database
 PAC7PD : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPD
 Tape is SYS012

.Output reports:
-Backup review SYS003

-Statistics on database SYS007

-Batch-procedure authorization option
 SYS005

. 0 : OK
. 12 : System error

DATABASE MANAGEMENT UTILITIES
 SAVE: DATABASE BACKUP
 SAVE: EXECUTION JCL

2
 2
 4

2.2.4. SAVE: EXECUTION JCL

```

===MOD SAVE
* $X JOB JNM=$PRFJ.SAVE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST3$PWL
/* *****
// JOB $PRFJ.SAVE          ***** $PROD $PREL *****
*
*          SEQUENTIAL BACKUP OF THE DATABASE
/* *****
/* INPUT FOR OPTIONAL REPORTS
/* COL 2      : 'OR'
/* COL 8      : ' ' PRINTING OF STATISTICS ABOUT DATABASE
/*           : 'N' NO STATISTICS PRINTING
/* COL 9      : ' ' PACBASE LIMITATIONS PRINTING
/*           : 'N' NO LIMITATIONS PRINTING
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
// ASSGN SYS003,$LST3 3TRD  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST00
* --- STEP 00 --- PTUBAS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTUBAS,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
            MODEL($MODEL'F) -
            REC (1 200) -
            ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
* $X SLI MEM=DF$PRO.$PFI.PC.SSLIT
* $X SLI MEM=DF$PRO.$PFI.PD.SSLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
OR
/*
/. JCLST03
* --- STEP 03 --- PTU500
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7PC ON TAPE : SYS015 ASSIGNED
// DLBL PAC7PD,'$PXU..SEQ.$PRO$PFI..NEWPD',,VSAM
/* PAC7PD ON TAPE : SYS012 ASSIGNED
// ON $RC GT 0 GOTO $EOJ
// EXEC PTU500,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04

```


DATABASE MANAGEMENT UTILITIES

SAVE: DATABASE BACKUP

2

2

SAVE: EXECUTION JCL

4

```
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PD.$SLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  DEL ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

| | | |
|---|------|----|
| | PAGE | 34 |
| DATABASE MANAGEMENT UTILITIES | | 2 |
| SASY: DATABASE SYSTEM BACKUP COMPLEMENT | | 3 |
| SASY: INTRODUCTION | | 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 | 35 |
| 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 | 2 |
| SASY: DATABASE SYSTEM BACKUP COMPLEMENT | 3 |
| SASY: DESCRIPTION OF STEPS | 2 |

2.3.2. SASY: DESCRIPTION OF STEPS

SASY: DESCRIPTION OF STEPS

DATABASE CONSISTENCY CHECK: PTUBAS

```
.Permanent input files:
-Data file
  PAC7AR : Physical name = $PXU..$PRO.$PFI.AR
-Error message file
  PAC7AE : Physical name = $PXY..$PRO.$PRO.AE

.Output report
-Validity report (Length=079)
                                     SYS005

. Return code :
. 0 : OK
. 12 : System error
```

BACKUP BY A SYSTEM UTILITY

The JCL in this step must be completed by the Database Manager.

SESSION NUMBER INCREMENTATION: PTU502

```
.Permanent input-output file:
-Data file
  PAC7AR : Physical name = $PXU..$PRO.$PFI.AR

.Permanent input file:
-Error message file
  PAC7AE : Physical name = $PXU..$PRO.$PRO.AE

.Output Report:
-Review
                                     SYS005
```

DATABASE MANAGEMENT UTILITIES

SASY: DATABASE SYSTEM BACKUP COMPLEMENT

2

3

SASY: EXECUTION JCL

3

2.3.3. SASY: EXECUTION JCL

```

===MOD SASY
* $X JOB JNM=$PRFJ.SASY,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.SASY          ***** $PROD $PREL *****
*          COMPLEMENT OF 'SYSTEM' BACK-UP OF THE DATABASE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- PTUBAS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTUBAS,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST02
* --- STEP 02 --- ??????
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
* =====
* INSERT HERE YOUR JCL FOR SYSTEM BACKUP
* =====
// EXEC ??????,SIZE=AUTO
/. JCLST03
* --- STEP 03 --- PTU502
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// EXEC PTU502,SIZE=AUTO
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|-------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 38 |
| REST: DATABASE RESTORATION | | 2 |
| REST: INTRODUCTION | | 4 |
| | | 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. !
+-----+
  
```

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 | 41 |
| 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 | 42 |
| 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.

2.4.3. REST: DESCRIPTION OF STEPS

REST: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

.Input file:
Card is SYS006
Tape is SYS006

.Output file:
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Permanent input file:
-Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report:
-Batch-procedure authorization option:
SYS005

.0 : OK
.8 : No authorization of batch procedure

VALIDATION OF JOURNAL EXISTENCE: IDCAMS

This step executes a LISTCAT on the Journal file (AJ).
Its return codes are:
0 : Journal file exist
4 : This file do not exist
(the control for archive is not executed).

VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed only if the Journal file exists.

.Permanent input files:
-Error message file
PAC7AJ : Physical name = \$PXU..\$PRO.\$PFI.AJ
-Journal file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report:
 SYS005
It is printed if the Journal file was not archived.

. 0 : Journal file has been archived
. 8 : Journal file has not been archived
(any step of REST/RESY is executed).

RESTORATION OF THE DATABASE: PTU400

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

.Permanent input files:
-Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
-Sequential image of the database
PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI..PC
Tape is SYS015
If backup option Dispatch:
-Sequential image of database %2
PAC7PD : Physical name = \$PXU..SEQ.\$PRO\$PFI..PD
Tape is SYS016

.Permanent output files:
-Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Index File
PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
-Journal file
PAC7AJ : Physical name = \$PXU..\$PRO.\$PFI.AJ

.Input transaction file:
-User transactions
PAC7MB : Physical name = \$XW..WREST.INPUT

.Output file:
-Working file (2 records)
PAC7PS : Physical name = \$XW..WREST.PS

.Output reports:
-Restoration report
SYS007
-Batch-procedure authorization option
SYS005

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 : Physical name = \$PXU..\$PRO.\$PFI.AR

.Permanent input files:

-Journal to apply

PAC7JO : Physical name = \$PXU..SEQ.\$PRO\$PFI..PJ

Tape is SYS011

-Error message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input work file:

PAC7PS : Physical name = \$XW..WREST.PS

.Output file:

-Update transactions

PAC7OJ : Physical name = \$XW..WREST.OJ

.Output report:

-Retrieval report

SYS005

. 0 : OK
. 12 : KO

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

DATABASE UPDATE: PACA15

.Permanent update files:

-Data file

ARLSR : Physical name = \$PXU..\$PRO.\$PFI.AR

-Index file

ANLSR : Physical name = \$PXU..\$PRO.\$PFI.AN

-Journal file

PAC7AJ : Physical name : \$PXU..\$PRO.\$PFI.AJ

.Permanent input files:

-Error message file

PAC7AE : Physical name : \$PXY..\$PRO.\$PRO.AE

-DSMS file of VA Pac elements

PAC7DC : Physical name = \$DSMS.DC

(DSM variant only)

.Input transaction file:

-Update transactions

PAC7MV : Physical name = \$XW..W\$MODUL..MV

.Output report(s):

-Update report

SYS005

-Erroneous-transaction list

SYS007

(The list of transactions belonging to a user is preceded by a banner specifying the user code.)

. Return code :

. 0 : OK
. 2 : Error 'Warning'
. 4 : Fatal error

DATABASE MANAGEMENT UTILITIES
 REST: DATABASE RESTORATION
 REST: EXECUTION JCL

2
 4
 4

2.4.4. REST: EXECUTION JCL

```

===MOD REST
* $X JOB JNM=$PRFJ.REST,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.REST          ***** $PROD $PREL *****
*          RELOADING-RESTORATION OF THE DATABASE
/* *****
/* INPUT
/* COL 2      : 'Y'
/* COL 3-7    : NUMBER OF GAPS IN ABSOLUTE VALUE
/* COL 8-9    : NUMBER OF GAPS AS A 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)
/* COL 13-15  : 'REC' FOR RECOVERY OF ARCHIVED TRANSACTIONS
/* COL 16-19  : 4 CHARACTERS DISPLAYED ON THE UPPER RIGHT
/*           : CORNER OF PACBASE SCREENS *USER.XXXX.LIB.SESS
/* COL 20-22  : MAXIMUM NUMBER OF ACCESSES IN SEQUENTIAL SEARCH
/*           : UNDER PAC-TP. DEFAULT IS 300.
/* COL 23     : U=IMPLICIT UPDATE (DEFAULT), N=EXPLICIT UPDATE
/* COL 28-33  : RESERVED FOR PACDESIGN PREFIXES. DO NOT USE.
/* COL 34     : NOT USED.
/* COL 35-46  : PF-KEYS CORRESPONDENCE TABLE
/* 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 NUMBE
/* OF EXISTING GAPS, AND OTHER DATA REMAINS UNCHANGED.
/*
/* IF THE JOURNAL FILE ON DISK (AJ) IS NOT REINITIALIZED,
/* THE RESTORATION PROCEDURE IS NOT EXECUTED.
/* IT IS THEREFORE NECESSARY TO EXECUTE THE ARCH PROCEDURE FIRST
/*
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
            MODEL($MODEL' ) -
            REC (1 200) -
            ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE OJ --- */
DEL          ($XW..W$MODUL..OJ) CL
DEF CL      (NAME ($XW..W$MODUL..OJ) -
            MODEL($MODEL' ) -
            REC (3000 500) -
            RECSZ(167 167) -
            RECFM(FB(167)) -
            ) DATA (NAME ($XW..W$MODUL..OJ.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PS --- */
DEL          ($XW..W$MODUL..PS) CL
DEF CL      (NAME ($XW..W$MODUL..PS) -
            MODEL($MODEL' ) -
            REC (2 1) -
            RECSZ(140 140) -
            RECFM(FB(140)) -
            ) DATA (NAME ($XW..W$MODUL..PS.D) )
/*

```

DATABASE MANAGEMENT UTILITIES

REST: DATABASE RESTORATION

2

4

REST: EXECUTION JCL

4

```

/. JCLST02
* --- STEP 02 --- PTU004
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU004,SIZE=AUTO
*USER      PASSWORD
Y      20F      $PRO.$PFI.
/*
// ON $RC<16 CONTINUE
/. JCLST03
/. JCLST03
// EXEC IDCAMS,SIZE=AUTO
// ON $RC GT 0 GOTO STEP05
// EXEC IDCAMS,SIZE=AUTO
LISTC ENT($PXU..$PRO.$PFI.AJ)
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTU380
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU380,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.AJ.$SLIT
* $X SLI MEM=DF$PRO.$PFI.AN.$SLIT
* $X SLI MEM=DF$PRO.$PFI.AR.$SLIT
/*
/. JCLST06
* --- STEP 06 --- PTU400
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..PC',,VSAM
/* PAC7PC ON TAPE : SYS015 ASSIGNED
// DLBL PAC7PD,'$PXU..SEQ.$PRO$PFI..PD',,VSAM
/* PAC7PD ON TAPE : SYS016 ASSIGNED
// DLBL PAC7PS,'$XW..W$MODUL..PS',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU400,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST07
* --- STEP 07 --- PTU420
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7JO,'$PXU..SEQ.$PRO$PFI..PJ',,VSAM
/* PAC7JO ON TAPE : SYS011 ASSIGNED
// DLBL PAC7OJ,'$XW..W$MODUL..OJ',,VSAM
// DLBL PAC7PS,'$XW..W$MODUL..PS',,VSAM,DISP=(,DELETE,KEEP)
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU420,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST08
* --- STEP 08 --- PAC15
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
===SEQ FOR DSM

```

DATABASE MANAGEMENT UTILITIES
REST: DATABASE RESTORATION
REST: EXECUTION JCL

2
4
4

```
// DLBL DSVCT, '$VCAD', ,VSAM
// DLBL PAC7DC, '$DSMS.DC', ,VSAM,CAT=DSVCT
===SEQ
// DLBL PAC7MV, '$XW..W$MODUL..OJ', ,VSAM,DISP=(,DELETE,KEEP)
// EXEC PACA15,SIZE=AUTO
/. STEPEND
* --- STEP 09 --- IDCAMS
// DLBL IJSYSUC, '$VCAP', ,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
DEL          ($XW..W$MODUL..OJ)    CL
DEL          ($XW..W$MODUL..PS)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```


| | | |
|--|------|----|
| | PAGE | 49 |
| 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 | 50 |
| DATABASE MANAGEMENT UTILITIES | | |
| RESY: DATABASE SYSTEM RESTORATION COMPLEMENT | | 2 |
| RESY: INTRODUCTION | | 5 |
| | | 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

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

RESY: USER INPUT - RESULTS

2

5

2

```

+-----+-----+-----+-----+
!POS.! LEN.! VALUE ! MEANING !
!-----!-----!-----!-----!
! 79 ! 1 ! 'D' ! 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 | 53 |
| 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

TRANSACTION RECOGNITION: PTU001

VALIDATION OF JOURNAL EXISTENCE: IDCAMS

This step executes a LISTCAT on the Journal file (AJ).
Its return codes are:
0 : Journal file exist
4 : This file do not exist
(the control for archive is not executed).

VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed only if the Journal file exists.

.Permanent input files:

-Error message file

PAC7AJ : Physical name = \$PXU..\$PRO.\$PFI.AJ

-Journal file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report:

SYS005

It is printed if the Journal file was not archived.

. 0 : Journal file has been archived
. 8 : Journal file has not been archived
(any step of REST/RESY is executed).

DEFINITION OF THE PACBASE DATABASE: IDCAMS

-Database files:

Files AR (data), AN (indices) and AJ (Journal) are defined according to the IDCAMS requests contained, respectively, in the following SLI members:

DF\$PRO.\$PFI.AR,DF\$PRO.\$PFI.AN,DF\$PRO.\$PFI.AJ

DATABASE MANAGEMENT UTILITIES

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

RESY: DESCRIPTION OF STEPS

2

5

3

DATABASE RESTORATION

The JCL in this step must be completed by the Database Manager.

DATABASE POSITIONING: PTU402

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

.Permanent output file:

-Data file

PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

.Permanent input file:

-Error message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input transaction file:

-User transaction

PAC7MB : Physical name = \$XW..WRESY.INPUT

. Output file:

-Work file (2 recs.)

PAC7PS : Physical name = \$XW..WRESY.PS

.Output report:

-Restoration report

SYS005

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 : Physical name = \$PXU..\$PRO.\$PFI.AR

.Permanent input files:

-Journal to apply

PAC7JO : Physical name = \$PXU..SEQ.\$PRO\$PFI..PJ

Tape is SYS011

-Error message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input work file:

PAC7PS : Physical name = \$XW..WREST.PS

.Output file:

-Update transactions

PAC7OJ : Physical name = \$XW..WREST.OJ

.Output report:

-Retrieval report

SYS005

. 0 : OK

. 12 : KO

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

DATABASE MANAGEMENT UTILITIES

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

RESY: DESCRIPTION OF STEPS

2

5

3

DATABASE UPDATE: PACA15

.Permanent update files:

-Data file

ARLSR : Physical name = \$PXU..\$PRO.\$PFI.AR

-Index file

ANLSR : Physical name = \$PXU..\$PRO.\$PFI.AN

-Journal file

PAC7AJ : Physical name : \$PXU..\$PRO.\$PFI.AJ

.Permanent input files:

-Error message file

PAC7AE : Physical name : \$PXY..\$PRO.\$PRO.AE

-DSMS file of VA Pac elements

PAC7DC : Physical name = \$DSMS.DC

(DSM variant only)

.Input transaction file:

-Update transactions

PAC7MV : Physical name = \$XW..W\$MODUL..MV

.Output report(s):

-Update report

SYS005

-Erroneous-transaction list

SYS007

(The list of transactions belonging to a user is preceded by a banner specifying the user code.)

. Return code :

. 0 : OK

. 2 : Error 'Warning'

. 4 : Fatal error

DATABASE MANAGEMENT UTILITIES

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

2

5

RESY: EXECUTION JCL

4

2.5.4. RESY: EXECUTION JCL

```

===MOD RESY
* $X JOB JNM=$PRFJ.RESY,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.RESY          ***** $PROD $PREL *****
*          'SYSTEM' RELOADING-RESTORATION COMPLEMENT
/* *****
/* INPUT
/* COL 2      : 'Y'
/* COL 3-7    : NUMBER OF GAPS IN ABSOLUTE VALUE
/* COL 8-9    : NUMBER OF GAPS AS A 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)
/* COL 13-15  : 'REC' FOR RECOVERY OF ARCHIVED TRANSACTIONS
/* COL 16-19  : 4 CHARACTERS DISPLAYED ON THE UPPER RIGHT
/*           : CORNER OF PACBASE SCREENS *USER.XXXX.LIB.SESS
/* COL 20-22  : MAXIMUM NUMBER OF ACCESSES IN SEQUENTIAL SEARCH
/*           : UNDER PAC-TP. DEFAULT IS 300.
/* COL 23     : U=IMPLICIT UPDATE (DEFAULT), N=EXPLICIT UPDATE
/* COL 28-33  : RESERVED FOR PACDESIGN PREFIXES. DO NOT USE.
/* COL 34     : NOT USED.
/* COL 35-46  : PF-KEYS CORRESPONDENCE TABLE
/* 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 NUMBE
/* OF EXISTING GAPS, AND OTHER DATA REMAINS UNCHANGED.
/*
/* IF THE JOURNAL FILE ON DISK (AJ) IS NOT REINITIALIZED,
/* THE RESTORATION PROCEDURE IS NOT EXECUTED.
/* IT IS THEREFORE NECESSARY TO EXECUTE THE ARCH PROCEDURE FIRST
/*
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE OJ --- */
DEL          ($XW..W$MODUL..OJ) CL
DEF CL       (NAME ($XW..W$MODUL..OJ) -
              MODEL($MODEL) -
              REC (3000 500) -
              RECSZ(165 165) -
              RECFM(FB(165)) -
              ) DATA (NAME ($XW..W$MODUL..OJ.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PS --- */
DEL          ($XW..W$MODUL..PS) CL
DEF CL       (NAME ($XW..W$MODUL..PS) -
              MODEL($MODEL) -
              REC (2 1) -
              RECSZ(140 140) -
              RECFM(FB(140)) -
              ) DATA (NAME ($XW..W$MODUL..PS.D) )
/*

```

DATABASE MANAGEMENT UTILITIES

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

2

5

RESY: EXECUTION JCL

4

```

/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER PASSWORD
Y 20F REC
/*
/. JCLST03
/. JCLST03
// EXEC IDCAMS,SIZE=AUTO
// ON $RC GT 0 GOTO STEP05
// EXEC IDCAMS,SIZE=AUTO
LISTC ENT($PXU..$PRO.$PFI.AJ)
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTU380
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTU380,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.AJ.$SLIT
* $X SLI MEM=DF$PRO.$PFI.AN.$SLIT
* $X SLI MEM=DF$PRO.$PFI.AR.$SLIT
/*
/. JCLST06
* --- STEP 06 --- ??????
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
* =====
* INSERT HERE YOUR JCL FOR SYSTEM RESTORATION
* =====
// EXEC ??????,SIZE=AUTO
/. JCLST07
* --- STEP 07 --- PTU402
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PS,'$XW..W$MODUL..PS',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU402,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST08
* --- STEP 08 --- PTU420
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7JO,'$PXU..SEQ.$PRO$PFI..PJ',,VSAM
/* PAC7JO ON TAPE : SYS011 ASSIGNED
// DLBL PAC7OJ,'$XW..W$MODUL..OJ',,VSAM
// DLBL PAC7PS,'$XW..W$MODUL..PS',,VSAM,DISP=(,DELETE,KEEP)
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU420,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST09
* --- STEP 09 --- PACA15
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM

```

DATABASE MANAGEMENT UTILITIES

RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

2

5

RESY: EXECUTION JCL

4

```
===SEQ FOR DSM
// DLBL DSVCT, '$VCAD', ,VSAM
// DLBL PAC7DC, '$DSMS.DC', ,VSAM,CAT=DSVCT
===SEQ
// DLBL PAC7MV, '$XW..W$MODUL..OJ', ,VSAM,DISP=(,DELETE,KEEP)
// EXEC PACA15,SIZE=AUTO
/. STEPEND
* --- STEP 10 --- IDCAMS
// DLBL IJSYSUC, '$VCAP', ,VSAM
// EXEC IDCAMS,SIZE=AUTO
  DEL          ($XW..W$MODUL..INPUT) CL
  DEL          ($XW..W$MODUL..OJ)    CL
  DEL          ($XW..W$MODUL..PS)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

| | | |
|-------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 60 |
| 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.

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 !ccyyymmdd! 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.)

| | | |
|---|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 62 |
| ARCH: JOURNAL ARCHIVAL | | 2 |
| ARCH: INPUT - RECOMMENDATIONS - RESULTS | | 6 |
| | | 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 'I' must be entered in column 15. As a result, the ARCH procedure formats a new Journal Back-up file.

| | PAGE | 63 |
|---|------|----|
| DATABASE MANAGEMENT UTILITIES | | |
| ARCH: JOURNAL ARCHIVAL | 2 | |
| ARCH: INPUT - RECOMMENDATIONS - RESULTS | 6 | |
| | 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.

2.6.3. ARCH: DESCRIPTION OF STEPS

ARCH: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

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 : Physical name = \$PXY..\$PRO.\$PRO.AE
-Previously archived transactions
PAC7JP : Physical name = \$PXU..SEQ.\$PRO\$PFI..PJ
Tape is SYS012
-Journal file to reinitialize
PAC7AJ : Physical name = \$PXU..\$PRO.\$PFI.AJ

.Input work file:

-User transaction
PAC7MB : Physical name = \$XW..WARCH.INPUT

.Permanent input-Output file:

-Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

.Output files:

-Archived update transactions
PAC7PJ : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPJ
Tape is SYS015
-Deactivated transactions
PAC7PQ : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPQ
Tape is SYS016
The DSN must be entered in order to keep these deactivated transactions.

.Output reports:

-Archival report
SYS007
-Batch-procedure authorization option
SYS005

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

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

PAGE

65

2
6
3

file's address.

.Input work file:

-User transaction

PAC7MB : Physical name = \$XW..WARCH.INPUT

.Permanent input/output file:

-Data file

PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

.Permanent input file:

-Error message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output file:

-Journal file to re-initialize

PAC7AJ : Physical name = \$PXU..\$PRO.\$PFI.AJ

.Output report:

-Review of reinitialization

SYS005

DATABASE MANAGEMENT UTILITIES
 ARCH: JOURNAL ARCHIVAL
 ARCH: EXECUTION JCL

2
 6
 4

2.6.4. ARCH: EXECUTION JCL

```

===MOD ARCH
* $X JOB JNM=$PRFJ.ARCH,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.ARCH          ***** $PROD $PREL *****
*
  ARCHIVAL OF THE TRANSACTION FILE
/* *****
/* INPUT      : COMMAND FOR DEACTIVATION OF ARCHIVED TRANSACTIONS
/* COL 2      : 'S'
/* COL 3 A 6  : SESSION NUMBER
/* COL 7 A 12 : DATE (YYMMDD)
/* COL 13     : ' ' PRESENCE OF ARCHIVED TRANSACTION FILE
/*           : 'I' ABSENCE OF ARCHIVED TRANSACTION FILE
/* COL 14     : ' ' PRESENCE OF DATA FILE (AR)
/*           : 'D' ABSENCE OF DATA FILE (AR)
/* COL 15     : ' ' ARCHIVAL AND REINITIALIZATION
/*           : 'J' REINITIALIZATION WITHOUT ARCHIVAL
/*           : WITHOUT THE LAST ARCHIVED MOVEMENTS
/* IN THE ABSENCE OF INPUT (OR AN 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 ON THE FILE OF DEACTIVATED TRANSACTIONS.
/*
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE BACKUP FILE PJ --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPJ) CL
DEF CL       (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ) -
              MODEL($MODEL) -
              REC ($RAJP $RAJS) -
              RECSZ(167 167) -
              RECFM(FB(167)) -
              ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ.D) )
/* --- DELETE-DEFINE BACKUP FILE PQ --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPQ) CL
DEF CL       (NAME ($PXU..SEQ.$PRO$PFI..NEWPQ) -
              MODEL($MODEL) -
              REC ($RAJP $RAJS) -
              RECSZ(167 167) -
              RECFM(FB(167)) -
              ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPQ.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
S          I
/*

```

DATABASE MANAGEMENT UTILITIES

ARCH: JOURNAL ARCHIVAL

2

6

ARCH: EXECUTION JCL

4

```

/. JCLST03
* --- STEP 03 --- PTU300
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7JP,'$PXU..SEQ.$PRO$PFI..PJ',,VSAM
/* PAC7JP ON TAPE : SYS012 ASSIGNED
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,KEEP)
// DLBL PAC7PJ,'$PXU..SEQ.$PRO$PFI..NEWPJ',,VSAM
/* PAC7PJ ON TAPE : SYS015 ASSIGNED
// DLBL PAC7PQ,'$PXU..SEQ.$PRO$PFI..NEWPQ',,VSAM
/* PAC7PQ ON TAPE : SYS016 ASSIGNED
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU300,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
/. JCLST04
// EXEC IDCAMS,SIZE=AUTO
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.AJ.$SLIT
/*
/. JCLST05
* --- STEP 05 --- PTU320
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU320,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST06
/. JCLST06
// EXEC IDCAMS,SIZE=AUTO
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PJ.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PQ.$SLIT
/*
/. STEPEND
* --- STEP 07 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|-------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 68 |
| REOR: DATABASE REORGANIZATION | | 2 |
| REOR: INTRODUCTION | | 7 |
| | | 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 | 69 |
| 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)

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.

| | | |
|-------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 72 |
| REOR: DATABASE REORGANIZATION | | 2 |
| REOR: INPUT - RECOMMENDATIONS | | 7 |
| | | 2 |

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

USER INPUTS CONTROL : PTU2CL

This step controls all user inputs and display a return code if there is an error.

.Permanent input files:
-Error message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input work file:
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:
-Formatted records
 PAC7BM : Physical name = \$XW..W\$MODUL..BM

.Output reports:
-Control report
 SYS005
-Batch-procedure authorization option
 SYS007

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 : Physical name = \$PXY..\$PRO.\$PRO.AE
-Sequential image of the database
 PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI.PC

.Output file:
-Formatted records
 PAC7MB : Physical name = \$XW..W\$MODUL.INPUT
 Tape is SYS014

.Output reports:
-Retrieval statistics
 SYS007

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:

.Input work file:

DATABASE MANAGEMENT UTILITIES

REOR: DATABASE REORGANIZATION

2

REOR: DESCRIPTION OF STEPS

7

3

```

-User transactions
  PAC7MB : Physical name = $XW..$WREOR.INPUT

.Permanent input file:
-Error messages
  PAC7AE : Physical name = $PXY..$PRO.$PRO.AE

.Output file:
-Entity records to purge
  PAC7PU : Physical name = $XW..$WREOR.PU

.Output report:
-Entity-purge transactions
                                         SYS005
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

.Input work files:
-Sorted records
  PAC7PR : Physical name = $XW..$WREOR.PR
-Entity records to be purged
  PAC7PU : Physical name = $XW..$WREOR.PU
-User transactions
  PAC7MB : Physical name = $XW..$WREOR.INPUT

.Permanent input file:
-Error message file
  PAC7AE : Physical name = $PXY..$PRO.$PRO.AE

.Output work files:
-Purged records
  PAC7QS : Physical name = $XW..WREOR.QS
  Tape is          SYS014
-Macro-Structure call lines
  PAC7UM : Physical name = $XW..WREOR.UM
  Tape is          SYS015

.Output reports:
-Library and session purge report
                                         SYS005
-Entity-purge report
                                         SYS007
-Technical report
                                         SYS003

```

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

INDEX RECONSTRUCTION: PTU220

This step executes two types of procedures:

```

.Reconstruction of the indexes using the data

.Separation of current and frozen sessions

```

```

.Input work files:
-Purged data
  PAC7UR : Physical name = $XW..WREOR.QS
  Tape is          SYS018
-Macro-Structure call lines
  PAC7UM : Physical name = $XW..WREOR.UM
  Tape is          SYS017

.Permanent input file:
-Error message file
  PAC7AE : Physical name = $PXY..$PRO.$PRO.AE

```

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

2
 7
 3

.Output files:
 -Data from frozen sessions
 PAC7PA : Physical name = \$XW..WREOR.PA
 Tape is SYS014
 -Data from the current session
 PAC7PB : Physical name = \$XW..WREOR.PB
 Tape is SYS015
 -First data record
 PAC7PC : Physical name = \$XW..WREOR.PC
 -Temporary index file
 PAC7AN : Physical name = \$XW..WREOR.AN

Tape is SYS013

.Work file (output, then input)
 -Macro-Structure call lines
 PAC7MR : Physical name = \$XW..WREOR.MR

.Output report:
 -Index-building report
 SYS005

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 : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input work files:
 -User transactions
 PAC7MB : Physical name = \$XW..WREOR.INPUT
 -Data from the frozen session
 PAC7PA : Physical name = \$XW..WREOR.PA
 Tape is SYS017
 -Data from the current session
 PAC7PB : Physical name = \$XW..WREOR.PB
 Tape is SYS018
 -First data record
 PAC7PC : Physical name = \$XW..WREOR.PC
 -Sorted index file
 PAC7AN : Physical name = \$XW..WREOR.AN

.Permanent output file:
 -Sequential image of the database
 PAC7CP : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPC
 Tape is SYS012
 If Dispatch option of backup:
 -Sequential image of the database ¾2
 PAC7PD : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPD
 Tape is SYS014

.Output report:
 -Logical database building
 SYS005

DATABASE MANAGEMENT UTILITIES
 REOR: DATABASE REORGANIZATION
 REOR: EXECUTION JCL

2
 7
 4

2.7.4. REOR: EXECUTION JCL

```

===MOD REOR
* $X JOB JNM=$PRFJ.REOR,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST3$PWL
/* *****
// JOB $PRFJ.REOR          ***** $PROD $PREL *****
*
*          SEQUENTIAL DATABASE REORGANIZATION
/* *****
/* WARNING : THE PROGRAM PTU220 NEEDS A PARTITION SIZE OF 900K
/*          PLUS A SUFFICIENT VIRTUAL STORAGE AMOUNT FOR
/*          THE CONCURRENT OPENING OF 8 VSAM FILES
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
// ASSGN SYS003,$LST3 3TRD PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
            MODEL($MODEL$) -
            REC (1 200) -
            ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE AN --- */
DEL          ($XW..W$MODUL..AN) CL
DEF CL      (NAME ($XW..W$MODUL..AN) -
            MODEL($MODEL$) -
            REC ($RPCP $RPCS) -
            RECSZ(55 55) -
            RECFM(FB(55)) -
            ) DATA (NAME ($XW..W$MODUL..AN.D) )
/* --- DELETE-DEFINE TEMPORARY FILE BM --- */
DEL          ($XW..W$MODUL..BM) CL
DEF CL      (NAME ($XW..W$MODUL..BM) -
            MODEL($MODEL$) -
            REC (200 200) -
            RECSZ(80 80) -
            RECFM(FB(80)) -
            ) DATA (NAME ($XW..W$MODUL..BM.D) )
/* --- DELETE-DEFINE TEMPORARY FILE MR --- */
DEL          ($XW..W$MODUL..MR) CL
DEF CL      (NAME ($XW..W$MODUL..MR) -
            MODEL($MODEL$) -
            REC ($RPCP $RPCS) -
            RECSZ(140 140) -
            RECFM(FB(140)) -
            ) DATA (NAME ($XW..W$MODUL..MR.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PA --- */
DEL          ($XW..W$MODUL..PA) CL
DEF CL      (NAME ($XW..W$MODUL..PA) -
            MODEL($MODEL$) -
            REC ($RPCP $RPCS) -
            RECSZ(149 149) -
            RECFM(FB(149)) -
            ) DATA (NAME ($XW..W$MODUL..PA.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PB --- */
DEL          ($XW..W$MODUL..PB) CL
DEF CL      (NAME ($XW..W$MODUL..PB) -
            MODEL($MODEL$) -
            REC ($RPCP $RPCS) -
            RECSZ(149 149) -
            RECFM(FB(149)) -

```

DATABASE MANAGEMENT UTILITIES
 REOR: DATABASE REORGANIZATION
 REOR: EXECUTION JCL

2
 7
 4

```

    ) DATA (NAME ($XW..W$MODUL..PB.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PC --- */
DEL ($XW..W$MODUL..PC) CL
DEF CL (NAME ($XW..W$MODUL..PC) -
MODEL($MODEL) -
REC (1 1) -
RECSZ(149 149) -
RECFM(FB(149)) -
) DATA (NAME ($XW..W$MODUL..PC.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PR --- */
DEL ($XW..W$MODUL..PR) CL
DEF CL (NAME ($XW..W$MODUL..PR) -
MODEL($MODEL) -
REC ($RPCP $RPCS) -
RECSZ(165 165) -
RECFM(FB(165)) -
) DATA (NAME ($XW..W$MODUL..PR.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PU --- */
DEL ($XW..W$MODUL..PU) CL
DEF CL (NAME ($XW..W$MODUL..PU) -
MODEL($MODEL) -
REC ($RPCP $RPCS) -
RECSZ(13 13) -
RECFM(FB(13)) -
) DATA (NAME ($XW..W$MODUL..PU.D) )
* $X SLI MEM=DF$PRO.$PFI.PC.$SLIT
* $X SLI MEM=DF$PRO.$PFI.PD.$SLIT
/* --- DELETE-DEFINE TEMPORARY FILE QS --- */
DEL ($XW..W$MODUL..QS) CL
DEF CL (NAME ($XW..W$MODUL..QS) -
MODEL($MODEL) -
REC ($RPCP $RPCS) -
RECSZ(165 165) -
RECFM(FB(165)) -
) DATA (NAME ($XW..W$MODUL..QS.D) )
/* --- DELETE-DEFINE TEMPORARY FILE UM --- */
DEL ($XW..W$MODUL..UM) CL
DEF CL (NAME ($XW..W$MODUL..UM) -
MODEL($MODEL) -
REC ($RPCP $RPCS) -
RECSZ(165 165) -
RECFM(FB(165)) -
) DATA (NAME ($XW..W$MODUL..UM.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PTU2CL
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7BM,'$XW..W$MODUL..BM',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU2CL,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTU200
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..PC',,VSAM
/* PAC7PC ON TAPE : SYS013 ASSIGNED
// DLBL PAC7PR,'$XW..W$MODUL..PR',,VSAM
/* PAC7PR ON TAPE : SYS014 ASSIGNED
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU200,SIZE=AUTO
/*

```

DATABASE MANAGEMENT UTILITIES
REOR: DATABASE REORGANIZATION
REOR: EXECUTION JCL

2
 7
 4

```

// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- SORT
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL SORTIN1,'$XW..W$MODUL..PR',,VSAM
// DLBL SORTOUT,'$XW..W$MODUL..PR',,VSAM,DISP=NEW
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 0 GOTO STEPEND
// EXEC SORT,SIZE=(AUTO,50K)
  SORT FIELDS=(1,21,CH,A,40,4,CH,A,28,4,CH,A,    $Y.X
              32,1,CH,D,46,4,CH,A,22,1,CH,D)
  RECORD TYPE=F,LENGTH=(165)
  INPFIL VSAM
  OUTFIL ESDS
/*
/*
// ON $RC<16 CONTINUE
/. JCLST06
* --- STEP 06 --- PTU208
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7PU,'$XW..W$MODUL..PU',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU208,SIZE=(AUTO,50K)
/*
/*
// ON $RC<16 CONTINUE
/. JCLST07
* --- STEP 07 --- PTU210
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7PR,'$XW..W$MODUL..PR',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7PR ON TAPE : SYS016 ASSIGNED
// DLBL PAC7PU,'$XW..W$MODUL..PU',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7QS,'$XW..W$MODUL..QS',,VSAM
/* PAC7QS ON TAPE : SYS014 ASSIGNED
// DLBL PAC7UM,'$XW..W$MODUL..UM',,VSAM
/* PAC7UM ON TAPE : SYS015 ASSIGNED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU210,SIZE=(AUTO,50K)
/*
/*
// ON $RC<16 CONTINUE
/. JCLST08
* --- STEP 08 --- PTU220
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$XW..W$MODUL..AN',,VSAM
/* PAC7AN ON TAPE : SYS013 ASSIGNED
// DLBL PAC7MR,'$XW..W$MODUL..MR',,VSAM
// DLBL PAC7PA,'$XW..W$MODUL..PA',,VSAM
/* PAC7PA ON TAPE : SYS014 ASSIGNED
// DLBL PAC7PB,'$XW..W$MODUL..PB',,VSAM
/* PAC7PB ON TAPE : SYS015 ASSIGNED
// DLBL PAC7PC,'$XW..W$MODUL..PC',,VSAM
// DLBL PAC7UM,'$XW..W$MODUL..UM',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7UM ON TAPE : SYS017 ASSIGNED
// DLBL PAC7UR,'$XW..W$MODUL..QS',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7UR ON TAPE : SYS018 ASSIGNED
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU220,SIZE=AUTO
/*
/*
// ON $RC<16 CONTINUE
/*
/. JCLST09
* --- STEP 09 --- SORT
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL SORTIN1,'$XW..W$MODUL..AN',,VSAM
// DLBL SORTOUT,'$XW..W$MODUL..AN',,VSAM,DISP=NEW
/* TAPES : LRECL 55, BLOCK 32725, FORMAT F, LABEL OMITTED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION

```

DATABASE MANAGEMENT UTILITIES
REOR: DATABASE REORGANIZATION
REOR: EXECUTION JCL

2
 7
 4

```

// ON $RC GT 0 GOTO STEPEND
// EXEC SORT,SIZE=(AUTO,50K)
SORT FIELDS=(1,55,CH,A)
RECORD TYPE=F,LENGTH=(55)
INPFIL VSAM
OUTFIL ESDS
/*
/*
// ON $RC<16 CONTINUE
/. JCLST10
* --- STEP 10 --- PTU240
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$XW..W$MODUL..AN',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7AN ON TAPE : SYS011 ASSIGNED
// DLBL PAC7CP,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7CP ON TAPE : SYS012 ASSIGNED
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PA,'$XW..W$MODUL..PA',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7PA ON TAPE : SYS017 ASSIGNED
// DLBL PAC7PB,'$XW..W$MODUL..PB',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7PB ON TAPE : SYS018 ASSIGNED
// DLBL PAC7PC,'$XW..W$MODUL..PC',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PD,'$PXU..SEQ.$PRO$PFI..NEWPD',,VSAM
/* PAC7PD ON TAPE : SYS014 ASSIGNED
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU240,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST11
* --- STEP 11 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PD.$SLIT
/*
/. STEPEND
* --- STEP 12 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
DEL ($XW..W$MODUL..AN) CL
DEL ($XW..W$MODUL..BM) CL
DEL ($XW..W$MODUL..MR) CL
DEL ($XW..W$MODUL..PA) CL
DEL ($XW..W$MODUL..PB) CL
DEL ($XW..W$MODUL..PC) CL
DEL ($XW..W$MODUL..PR) CL
DEL ($XW..W$MODUL..PU) CL
DEL ($XW..W$MODUL..QS) CL
DEL ($XW..W$MODUL..UM) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|---------------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 80 |
| 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

PAGE

81

2
8
2

2.8.2. SVAG: DESCRIPTION OF STEPS

SVAG: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

BACKUP OF GENERATION-PRINTING REQUESTS: PTU550

.Input files:

-Requests

PAC7AG : Physical name = \$PXU..\$PRO.\$PFI.AG

-Error messages

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

-User input

PAC7MB : Physical name = \$XW..WSVAG.INPUT

.Output file:

-Sequential image of requests

PAC7PG : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPG

Tape is SYS014

.Output reports:

-Backup report

SYS007

-Check on procedure-access authorization

SYS005

DATABASE MANAGEMENT UTILITIES
 SVAG: GENERATION-PRINT REQUEST BACKUP
 SVAG: EXECUTION JCL

2
 8
 3

2.8.3. SVAG: EXECUTION JCL

```

===MOD SVAG
* $X JOB JNM=$PRFJ.SVAG,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.SVAG ***** $PROD $PREL *****
* --- BACKUP OF "AG" FILE --- *
/* *****
/* NO INPUT *
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL ($XW..W$MODUL..INPUT) CL
DEF CL (NAME ($XW..W$MODUL..INPUT) -
MODEL($MODEL) -
REC (1 200) -
) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE BACKUP FILE PG --- */
DEL ($PXU..SEQ.$PRO$PFI..NEWPG) CL
DEF CL (NAME ($PXU..SEQ.$PRO$PFI..NEWPG) -
MODEL($MODEL) -
REC (100 100) -
RECSZ(150 150) -
RECFM(FB(150)) -
) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPG.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PTU550
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AG,'$PXU..$PRO.$PFI.AG',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PG,'$PXU..SEQ.$PRO$PFI..NEWPG',,VSAM
/* PAC7PG ON TAPE : SYS014 ASSIGNED
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU550,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PG.$SLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|--|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 83 |
| 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 ! | !uuuuuuuu! | ! User to be purged | ('AU')! |

Maximum number of sessions.....: 500

Maximum number of libraries.....: 100

Maximum number of users.....: 100

Default option: restoration.

2.9.3. REAG: DESCRIPTION OF STEPS

REAG: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

.Input file:
Card is SYS006
Tape is SYS006

.Output file:
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Permanent input file:
-Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report:
-Batch-procedure authorization option:
SYS005

.0 : OK
.8 : No authorization of batch procedure

INITIALIZATION-REORGANIZATION OF REQUEST FILE (AG): PTU560

.Permanent input files:
-Sequential image of requests
PAC7PG : Physical name = \$PXU..\$SEQ.\$PRO\$PFI..PG
Tape is SYS014
-Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Permanent output file:
-Request file
PAC7AG : Physical name = \$PXU..\$PRO.\$PFI.AG

.Input transaction file:
-User transactions
PAC7MB : Physical name = \$XW..WREAG.INPUT

.Output reports:
-Restoration report SYS007
-List of transactions SYS003
-Batch-procedure authorization option
 SYS005

DATABASE MANAGEMENT UTILITIES

REAG: GENERATION-PRINT REQUEST RESTORATION

REAG: EXECUTION JCL

2

9

4

2.9.4. REAG: EXECUTION JCL

```

===MOD REAG
* $X JOB JNM=$PRFJ.REAG,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST3$PWL
/* *****
// JOB $PRFJ.REAG          ***** $PROD $PREL *****
*           INITIALIZATION-RESTORATION AG FILE
/* *****
/* 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 LIBRARY XXX'S COMMANDS
/* COL 2-7   : 'ASXXX' DELETION OF SESSION XXXX'S COMMANDS
/* COL 2-11  : 'AUXXXXXXXX' DELETION OF USER XXXXXXXXX'S COMMANDS
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
// ASSGN SYS003,$LST3 3TRD  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU004
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU004,SIZE=AUTO
  AGI
/*
// ON $RC<16 CONTINUE
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.AG.$SLIT
/*
/. JCLST04
* --- STEP 04 --- PTU560
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AG,'$PXU..$PRO.$PFI.AG',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7PG,'$PXU..SEQ.$PRO$PFI..PG',,VSAM
/* PAC7PG ON TAPE : SYS014 ASSIGNED
// EXEC PTU560,SIZE=AUTO
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|---------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 87 |
| PARM: UPDATE OF USER PARAMETERS | | 2 |
| PARM: INTRODUCTION | | 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.

| | | |
|----------------------------------|------|----|
| | PAGE | 88 |
| 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

PAGE

89

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.

| | | |
|----------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 91 |
| PARM: UPDATE OF USER PARAMETERS | | 2 |
| PARM: INPUT - RECOMMENDATIONS | | 10 |
| | | 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 AEO, 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 AEO.

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.

DATABASE MANAGEMENT UTILITIES

PARM: UPDATE OF USER PARAMETERS

2

PARM: INPUT - RECOMMENDATIONS

10

2

2. USER PARAMETERS2.1 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

PARM: INPUT - RECOMMENDATIONS

10

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 !
-----!

```

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 | 95 |
| 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

The global authorization allows access to the entire database BUT the libraries

| | PAGE | 96 |
|---------------------------------|------|----|
| DATABASE MANAGEMENT UTILITIES | | 2 |
| PARM: UPDATE OF USER PARAMETERS | | 10 |
| PARM: USER-CODE DEFINITION | | 3 |

explicitly mentionned.

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

NOTES:

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

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

Example:

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

- . '1' in the GLOBAL AUTHORIZATION level or the DATABASE AUTHORIZATION level,
- . '3' in the LIBRARY AUTHORIZATION specific to 'AP1'.

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

Update of a library-authorization level

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

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

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

| | PAGE | 97 |
|--|------|----|
| 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.

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>No global access authorization.</p> <p>No global access authorization.</p> <p>Read-only access authorized for both current and all frozen sessions.</p> <p>Read-write access authorized for the current session and read-only access for all frozen sessions.</p> <p>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 Library Definition screen of the libraries concerned.</p> |

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

DATABASE MANAGEMENT UTILITIES

2

PARM: UPDATE OF USER PARAMETERS

10

PARM: USER-CODE SPECIFIC AUTHORIZATIONS

5

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> <p>Read-only access authorization on the whole database</p> |

DATABASE MANAGEMENT UTILITIES

2

PARM: UPDATE OF USER PARAMETERS

10

PARM: USER-CODE SPECIFIC AUTHORIZATIONS

5

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

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

PAGE

102

2
10
5

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

| | | |
|----------------------------------|------|-----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 103 |
| 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 | TYPE OF TEXT (REQ. IN CREATION) The TYPE OF TEXT field is used for documentation purposes only, and allows the user to: .obtain the list of texts sorted by type (CHOICE: LTT), .have explicit titles including the labels corresponding to the chosen type of text, on screens and reports which contain the text. The coding of types and labels depends on an external parameter handled by the Database Administrator. Default value. |
| 3 | 15 | | NAME OF TEXT TYPE (REQ. IN CREATION) Specify the label to appear with the corresponding Type of Text. NOTE: This label will appear on the Text Definition screen when the corresponding Type of Text is used, and on screens and reports which contain the text. Enter the name to appear with the corresponding Type of Text. This name will appear on the Text Definition screen when the corresponding Type of Text is used. |

| | | |
|--|------|-----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 105 |
| PARM: UPDATE OF USER PARAMETERS | | 2 |
| PARM: MODIFICATION OF STANDARD ERROR MESSAGES | | 10 |
| | | 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' |
| | | ABSC | To modify 'ABSENCE OF RECORD' |

| NUM | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|----------------|--|
| 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> |

| | PAGE | 108 |
|---|------|-----|
| DATABASE MANAGEMENT UTILITIES | | 2 |
| PARM: UPDATE OF USER PARAMETERS | | 10 |
| PARM: GENERATED-STREAM CONTROL CARDS | | 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.

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

| | PAGE | 109 |
|---|------|-----|
| DATABASE MANAGEMENT UTILITIES | | 2 |
| PARM: UPDATE OF USER PARAMETERS | | 10 |
| PARM: GENERATED-STREAM CONTROL CARDS | | 8 |

(they will be the default options for all programs in that library),

- . Enter the program 'front/back' options on the Program Definition screen if the default options are not appropriate,
- . Enter on-line program- and map- 'front/back' options on the Screen Definition screen,
- . Enter data-block 'front/back' options on the Database Block Definition screen.

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

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

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

PARAMETERIZATION OF CONTROL CARDS

Job control cards are parameterized according to the following principles:

A control card consists of three types of information:

- . A fixed part, representing the syntax of the job control language in use,
- . A first variable part, made up of components that can be determined in advance (such as the generated program code or the library name),
- . A second variable part, made up of fields that can be entered only at the last minute, because they depend on the run to be executed. (For example, SYSOUT class and time limit.)

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

It is specified in the line's last character.

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

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

- . Alphabetic parameters whose values are given in the input descriptions will be decoded in terms of their pre-established meaning.
- . Numeric parameters introduced on the screen or in the generation-print request transaction are decoded in terms of their user-specified meaning.

| | | |
|---------------------------------------|------|-----|
| DATABASE MANAGEMENT UTILITIES | PAGE | 110 |
| PARM: UPDATE OF USER PARAMETERS | | 2 |
| PARM: GENERATED-STREAM CONTROL CARDS | | 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 | | TYPE OF OPTION (REQ. IN CREATION) |
| | | A | Beginning of generated program job stream. |
| | | D | Before the generated program. |
| | | F | Following the generated program. |
| | | Z | Following the generated program job stream. |
| 2 | 1 | | OPTION CODE (REQ. IN CREATION) |
| | | | Identifies optional job control cards. |
| | | | To be specified for: |
| | | | - The 'Front/Back' of the job stream on the Library Definition screen, |
| | | | - The 'Front/Back' program options on the Library Definition screen or the Program Definition screen, |
| | | | - The 'Front/Back' options for the on-line program and for the map on the Screen Definition screen, |
| | | | - The 'Front/Back' block options on the Block Definition screen. |
| 3 | 2 | | LINE NUMBER (REQ. IN CREATION) |
| | | BLANK | Option title line: |
| | | 0 - 99 | Title in the "Optional Card Image" field. Lower-case keying accepted. |
| | | NUMERICAL | Optional control card: |
| | | | It is recommended to leave gaps in a line's number sequence in order to make future insertions possible. |
| 4 | 67 | | OPTIONAL CONTROL CARD IMAGE |
| | | | The image of the optional control card is written in compressed format. Parameterized information is represented by the INSERTION REFERENCE CHARACTER(S). |
| | | | The last column of this field (67th) is specified with the label "C". Any value other than blank entered in this column will be generated in column 72 of the control card. |
| | | | This field accepts lowercase characters. |
| | | | INPUT PARAMETERS |
| | | | Each of these parameters selects a data element from the internal or source system library: |
| | | A | Library code (*' entity, 1 to 3 characters). |

| NUM | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|-------------|---|
| | | B | Source library name (*' entity, 1-36 characters). |
| | | C | Current date including century (10 characters). |
| | | D | Current date determined by the system, in eight-character format. |
| | | G | Session number of the database when the job runs (5 characters). |
| | | I | DSMS change number |
| | | J | Name of the job initialized by the System (IMS only). |
| | | K | No. of the job initialized by the System (IMS only). |
| | | L | Parameter required for operation of the VA Pac-Endevor Interface. It may also be used to suit user needs. Its purpose is to select the data provided by Pacbase Constants, in the following format: EEntityNomexterBasBibSessTjj/mm/aahh:mm:ssUserCode With: E (1) = Entity type (O, M for Map, P, or B) Entity (6) = VisualAge Pacbase Entity code Nomexter (8) = External name Base (4) = Database code Bib (3) = Library code Sess (4) = Generation session number T (1) = Session status (T or blank) 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. |
| | | V | Job stream number (two-digit value), automatically assigned according to the order of execution. |

DATABASE MANAGEMENT UTILITIES

PARM: UPDATE OF USER PARAMETERS

2

10

PARM: GENERATED-STREAM CONTROL CARDS

8

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

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

PAGE

114

2
 10
 8

| NUM | LEN | CLASS VALUE | DESCRIPTION OF FIELDS AND FILLING MODE |
|-----|-----|----------------|--|
| | | | <p>meter.</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> |

2.10.9. PARM: DESCRIPTION OF STEPS

PARM: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

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 : Physical name = \$PXY..\$PRO.\$PRO.AE

-User parameters

PAC7AP : Physical name = \$PXY..\$PRO.\$PRO.AP

.Permanent input files:

-User parameter backup

PAC7EC : Physical name = \$PXY..SEQ.\$PRO\$PRO.PE

.Transaction file:

-Update transactions

PAC7MC : Physical name = \$XW..WPARM.INPUT

.Output file

-User parameter backup

PAC7CE : Physical name = \$PXY..SEQ.\$PRO\$PRO.NEWPE

.Output reports

-Printing of the update file and review

SYS007

-Check on procedure access authorization

SYS005

.Return codes:

. 0 : OK and file reloading of AE and AP

. 4 : OK and not reloading of AE and AP

. 8 : No authorization about parameter updating

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 : Physical name = \$PXY..SEQ.\$PRO\$PRO.NEWPE

-Initial sequential image of

error messages

PAC7LE : Physical name = \$PXY..SEQ.\$PRO\$PRO.AE0

.Transaction file:

-Update transactions

PAC7MB : Physical name = \$XW.WPARM.INPUT

DATABASE MANAGEMENT UTILITIES

PARM: UPDATE OF USER PARAMETERS

2

PARM: DESCRIPTION OF STEPS

10

9

.Permanent output files:

-Error messages to be rebuilt

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

-User parameters to be recreated

PAC7AP : Physical name = \$PXY..\$PRO.\$PRO.AP

.Output report:

-Reconstruction report

SYS005

.Sort file(s):

DATABASE MANAGEMENT UTILITIES

PARAM: UPDATE OF USER PARAMETERS

2

10

PARAM: EXECUTION JCL

10

2.10.10. PARAM: EXECUTION JCL

```

===MOD PARM
* $X JOB JNM=$PRFJ.PARM,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.PARM ***** $PROD $PREL *****
*
  UPDATE OF USER PARAMETERS
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
  DEL          ($XW..W$MODUL..INPUT) CL
  DEF CL       (NAME ($XW..W$MODUL..INPUT) -
                MODEL($MODEL) -
                REC (1 200) -
                ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
  /* --- DELETE-DEFINE BACKUP FILE PE --- */
  DEL          ($PX..SEQ.$PRO$PRO..NEWPE) CL
  DEF CL       (NAME ($PX..SEQ.$PRO$PRO..NEWPE) -
                MODEL($MODEL) -
                REC (1000 1000) -
                RECSZ(80 80) -
                RECFM(FB(80)) -
                ) DATA (NAME ($PX..SEQ.$PRO$PRO..NEWPE.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
  *****
  NRCHAR
XNK000<----- SYSTEM ACCESS KEYS ----->$K
  NUTEST      000          44
  NUTEST      010          $E 3
  NUXXXXXXXXX000      1
  NUXXXXXXXXX010$PRO$PFI
/*
/. JCLST03
* --- STEP 03 --- PACU15
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AP,'$PX..$PRO.$PRO.AP',,VSAM
// DLBL PAC7CE,'$PX..SEQ.$PRO$PRO..NEWPE',,VSAM
// DLBL PAC7EC,'$PX..SEQ.$PRO$PRO..PE',,VSAM
// DLBL PAC7MC,'$XW..W$MODUL..INPUT',,VSAM
// ON $RC EQ 4 GOTO STEP05
// ON $RC GT 4 GOTO $EOJ
// EXEC PACU15,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PACU80
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM,DISP=NEW
// DLBL PAC7AP,'$PX..$PRO.$PRO.AP',,VSAM,DISP=NEW
// DLBL PAC7CE,'$PX..SEQ.$PRO$PRO..NEWPE',,VSAM
// DLBL PAC7LE,'$PX..SEQ.$PRO$PRO..AE',,VSAM
// DLBL PAC7MC,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PACU80,SIZE=AUTO
/. JCLST05

```

DATABASE MANAGEMENT UTILITIES

PARM: UPDATE OF USER PARAMETERS

PARM: EXECUTION JCL

2

10

10

```
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PRO.PE.$SLIT
/*
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

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

PAGE 119

3

3. VERSIONING UTILITIES

3.1. PEI: PRODUCTION ENVIRONMENT INTERFACE

3.1.1. PEI: OVERVIEW

PEI: INTRODUCTION

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

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

- . Management of all GENERATION ENVIRONMENTS defined on-site (production, system acceptance, test, etc.);
- . Follow-up of entities generated from the database and managed in any on-site environment;
- . Automatic session freeze when needed (for example, when generating into a production environment);
- . The possibility to manually request a session freeze;
- . Generation of purge requests for redundant frozen sessions;
- . A list of frozen sessions for which there were entities put into production;
- . Information related to these entities, such as the library code, the code of the user, and the session number of the last generation and of the most recent database freeze;
- . Project(s) follow-up by development team(s) in relation to generated entities.

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

| | | |
|---------------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 121 |
| PEI: PRODUCTION ENVIRONMENT INTERFACE | | 3 |
| PEI: OVERVIEW | | 1 |
| | | 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.

| | | |
|---------------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 122 |
| PEI: PRODUCTION ENVIRONMENT INTERFACE | | 3 |
| INPE: FILE INITIALIZATION | | 1 |
| | | 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

123

3
1
2

3.1.2.2. INPE: DESCRIPTION OF STEPS

INPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

PEI INITIAL BACKUP: PACR01

.Permanent input files:
-Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Index file
PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
-Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input file:
-User input file
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:
-PEI initial backup
PAC7PP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPP

.Output reports:
-Execution report
SYS007
-Batch-procedure authorization option
SYS005

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

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

INPE: FILE INITIALIZATION

3

1

2

3.1.2.3. INPE: EXECUTION JCL

```

===SEQ FOR PEI
===MOD INPE
* $X JOB JNM=$PRFJ.INPE,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.INPE          ***** $PROD $PREL *****
/*   PEI OPTION           BACK-UP FILE INITIALIZATION
/* *****
/*   FOLLOW BY RESTORATION PROCEDURE RSPE EXECUTION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE BACKUP FILE PP --- */
DEL          ($PRO..SEQ.$PRO$PFI..NEWPP) CL
DEF CL       (NAME ($PRO..SEQ.$PRO$PFI..NEWPP) -
              MODEL($MODEL) -
              REC (200 200) -
              RECSZ(110 110) -
              RECFM(FB(110)) -
              ) DATA (NAME ($PRO..SEQ.$PRO$PFI..NEWPP.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PACR01
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PP,'$PXU..SEQ.$PRO$PFI..NEWPP',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 4 GOTO STEPEND
// EXEC PACR01,SIZE=(AUTO,50K)
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PP.SSLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*

```

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

INPE: FILE INITIALIZATION

3

1

2

```
// EXEC LISTLOG
/ &
* $X EOJ
===SEQ
```

| | | |
|---------------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 126 |
| 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

127

3
1
3

3.1.3.2. SVPE: DESCRIPTION OF STEPS

SVPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

PEI BACKUP: PACR60

.Permanent input files:
- 'Batch' PEI file
PAC7AB : Physical name = \$PXU..\$PRO.\$PFI.AB
- 'On-line' PEI file
PAC7AC : Physical name = \$PXU..\$PRO.\$PFI.AC
- Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
- Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output file:
- PEI backup
PAC7PP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPP
Tape is SYS015

.Input file:
- Transaction file
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output reports:
- Execution report
SYS007
- Batch-procedure authorization option
SYS005

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

SVPE: FILE BACKUP

3

1

3

3.1.3.3. SVPE: EXECUTION JCL

```

===SEQ FOR PEI
===MOD SVPE
* $X JOB JNM=$PRFJ.SVPE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.SVPE          ***** $PROD $PREL *****
*      MODULE PEI : SAUVEGARDE DES FICHIERS
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE BACKUP FILE PP --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPP) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWPP) -
             MODEL($MODEL) -
             REC (200 200) -
             RECSZ(110 110) -
             RECFM(FB(110)) -
             ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPP.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PACR60
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AB,'$PXU..$PRO.$PFI.AB',,VSAM
// DLBL PAC7AC,'$PXU..$PRO.$PFI.AC',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PP,'$PXU..SEQ.$PRO$PFI..NEWPP',,VSAM
/* PAC7PP ON TAPE : SYS015 ASSIGNED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 4 GOTO STEPEND
// EXEC PACR60,SIZE=(AUTO,50K)
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PP.SSLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*

```


VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

SVPE: FILE BACKUP

3

1

3

```
// EXEC LISTLOG
/ &
* $X EOJ
===SEQ
```

| | |
|---------------------------------------|--|
| VERSIONING UTILITIES | |
| PEI: PRODUCTION ENVIRONMENT INTERFACE | |
| RSPE: FILE RESTORATION | |

| |
|---|
| 3 |
| 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

131

3
1
4

3.1.4.2. RSPE: DESCRIPTION OF STEPS

RSPE: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

.Input file:
Card is SYS006
Tape is SYS006

.Output file:
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Permanent input file:
-Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report:
-Batch-procedure authorization option:
SYS005

.0 : OK
.8 : No authorization of batch procedure

PEI RESTORATION: PACR61

.Input file:
-User input
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Permanent input files:
-Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
-Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-PEI backup file
PAC7PP : Physical name = \$PXU..SEQ.\$PRO.\$PFI.PP
Tape is SYS015

.Permanent output files:
-'Batch' PEI file
PAC7AB : Physical name = \$PXU..\$PRO.\$PFI.AB
-'On-line' PEI file
PAC7AC : Physical name = \$PXU..\$PRO.\$PFI.AC

.Output reports:
-Review SYS007
-Batch-procedure authorization option SYS005

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

RSPE: FILE RESTORATION

3

1

4

3.1.4.3. RSPE: EXECUTION JCL

```

===SEQ FOR PEI
===MOD RSPE
* $X JOB JNM=$PRFJ.RSPE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.RSPE          ***** $PROD $PREL *****
*      PEI OPTION : FILE RESTORATION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
            MODEL($MODEL) -
            REC (1 200) -
            ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU004
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU004,SIZE=AUTO
*USER      PASSWORD
/. JCLST03
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.AB.SSLIT
* $X SLI MEM=DF$PRO.$PFI.AC.SSLIT
/*
/. JCLST04
* --- STEP 04 --- PACR61
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AB,'$PXU..$PRO.$PFI.AB',,VSAM
// DLBL PAC7AC,'$PXU..$PRO.$PFI.AC',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PP,'$PXU..SEQ.$PRO$PFI..PP',,VSAM
/* PAC7PP ON TAPE : SYS015 ASSIGNED
// EXEC PACR61,SIZE=AUTO
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

| | | |
|--|------|-----|
| VERSIONING UTILITIES | PAGE | 133 |
| PEI: PRODUCTION ENVIRONMENT INTERFACE | | 3 |
| PRPE: PRODUCTION ENVIRONMENT PRINTOUTS | | 1 |
| | | 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

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

135

3
1
5

3.1.5.3. PRPE: DESCRIPTION OF STEPS

PRPE: DESCRIPTION OF STEPS

PEI PRINTING: PACR10

.Permanent input files:
- 'Batch' PEI file
 PAC7AB : Physical name = \$PXU..\$PRO.\$PFI.AB
- 'On-line' PEI file
 PAC7AC : Physical name = \$PXU..\$PRO.\$PFI.AC
- Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
- Index file
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
- Error-message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input transaction file:
- Printing requests
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output reports:
- Printouts
 SYS007
- Batch-procedure authorization option
 SYS005

.Sort file(s):

.Return code(s):

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

PRPE: PRODUCTION ENVIRONMENT PRINTOUTS

3

1

5

3.1.5.4. PRPE: EXECUTION JCL

```

===SEQ FOR PEI
===MOD PRPE
* $X JOB JNM=$PRFJ.PRPE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.PRPE          ***** $PROD $PREL *****
*
*          PEI OPTION
*
*          PRODUCTION ENVIRONMENT PRINTING
/* *****
/* INPUT : PRINTING REQUEST
/* COL 2-3 : 'PL' (CARD CODE)
/* ***** TYPES OF LIST *****
/* COL 4   : '1' ENVIRONMENT 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 ENTITY CODE
/* COL 9   : '1' ENVIRONMENTS PER EXTERNAL NAME
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL$) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
PL111111
/*
/. JCLST03
* --- STEP 03 --- PACR10
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AB,'$PXU..$PRO.$PFI.AB',,VSAM
// DLBL PAC7AC,'$PXU..$PRO.$PFI.AC',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// EXEC PROC=$SORTWK CALL SORTWK DEFINITION
// EXEC PACR10,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```


| | | |
|---|------|-----|
| VERSIONING UTILITIES | PAGE | 137 |
| PEI: PRODUCTION ENVIRONMENT INTERFACE | | 3 |
| GRPE: TRANSACTION-GENERATION FOR REORGANIZATION | | 1 |
| | | 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

138

3
1
6

3.1.6.2. GRPE: DESCRIPTION OF STEPS

GRPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

GENERATION OF TRANSACTIONS FOR REORGANIZATION: PACR40

.Permanent input files:
- 'Batch' PEI file
 PAC7AB : Physical name = \$PXU..\$PRO.\$PFI.AB
- 'On-line' PEI file
 PAC7AC : Physical name = \$PXU..\$PRO.\$PFI.AC
- Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
- Index file
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
- Error message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input file:
- User input
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:
- Generated trans. for reorganization

.Output reports:
- Execution report

- Batch-procedure authorization option

.Sort file(s):

.Return code(s):

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

GRPE: TRANSACTION-GENERATION FOR REORGANIZATION

3

1

6

3.1.6.3. GRPE: EXECUTION JCL

```

===SEQ FOR PEI
===MOD GRPE
* $X JOB JNM=$PRFJ.GRPE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.GRPE          ***** $PROD $PREL *****
*      PEI OPTION : GENERATION OF TRANSACTIONS FOR          *
*      REORGANIZING THE PACBASE DATA BASE                  *
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
            MODEL($MODEL' ) -
            REC (1 200) -
            ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE RESULT FILE MV --- */
DEL          ($PXU..R$MODUL..MV) CL
DEF CL      (NAME ($PXU..R$MODUL..MV) -
            MODEL($MODEL' ) -
            REC (1000 1000) -
            RECSZ(80 80) -
            RECFM(FB(80)) -
            ) DATA (NAME ($PXU..R$MODUL..MV.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PACR40
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AB,'$PXU..$PRO.$PFI.AB',,VSAM
// DLBL PAC7AC,'$PXU..$PRO.$PFI.AC',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7MV,'$PXU..R$MODUL..MV',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PACR40,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

| | | |
|---------------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 140 |
| PEI: PRODUCTION ENVIRONMENT INTERFACE | | 3 |
| HIPE: AUTOMATIC SESSION FREEZE | | 1 |
| | | 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

3
1
7

3.1.7.3. HIPE: DESCRIPTION OF STEPS

HIPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

DATABASE CONSISTENCY CHECK: PTUBAS

.Permanent input files:
-Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Error message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report
-Validity report (Length=079)
 SYS005

. Return code :
 . 0 : OK
 . 12 : System error

AUTOMATIC SESSION FREEZE: PACR30

.Permanent input files:
-'Batch' PEI file
 PAC7AB : Physical name = \$PXU..\$PRO.\$PFI.AB
-'On-line' PEI file
 PAC7AC : Physical name = \$PXU..\$PRO.\$PFI.AC
-Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Index file
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
-Journal file
 PAC7AJ : Physical name = \$PXU..\$PRO.\$PFI.AJ
-Error message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input transaction file:
-Session freeze requests
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output report:
-Execution report
 SYS005

.Work files:
 PAC7MW : Physical name = \$XW..W\$MODUL..MW
 PAC7WB : Physical name = \$XW..W\$MODUL..WB

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

HIPE: AUTOMATIC SESSION FREEZE

3

1

7

3.1.7.4. HIPE: EXECUTION JCL

```

===SEQ FOR PEI
===MOD HIPE
* $X JOB JNM=$PRFJ.HIPE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.HIPE          ***** $PROD $PREL *****
*   PEI OPTION :    AUTOMATIC SESSION FREEZE
/* *****
/* INPUT : REQUIRED '*' LINE
/*           OPTIONAL 'X1' LINE (AUTOMATICALLY GENERATED)
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
/* GOTO JCLSTXX
/. JCLST00
* --- STEP 00 --- PTUBAS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// ON $RC GT 0 GOTO $EOJ
// EXEC PTUBAS,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE MW --- */
DEL          ($XW..W$MODUL..MW) CL
DEF CL       (NAME ($XW..W$MODUL..MW) -
              MODEL($MODEL) -
              REC (1000 1000) -
              RECSZ(167 167) -
              RECFM(FB(167)) -
              ) DATA (NAME ($XW..W$MODUL..MW.D) )
/* --- DELETE-DEFINE TEMPORARY FILE WB --- */
DEL          ($XW..W$MODUL..WB) CL
DEF CL       (NAME ($XW..W$MODUL..WB) -
              MODEL($MODEL) -
              REC (1000 1000) -
              RECSZ(110 110) -
              RECFM(FB(110)) -
              ) DATA (NAME ($XW..W$MODUL..WB.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER  PASSWORD***
/*
/. JCLST03
* --- STEP 03 --- PACR30
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AB,'$PXU..$PRO.$PFI.AB',,VSAM
// DLBL PAC7AC,'$PXU..$PRO.$PFI.AC',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7MW,'$XW..W$MODUL..MW',,VSAM
// DLBL PAC7WB,'$XW..W$MODUL..WB',,VSAM

```

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

HIPE: AUTOMATIC SESSION FREEZE

3

1

7

```
// EXEC PACR30,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  DEL          ($XW..W$MODUL..INPUT) CL
  DEL          ($XW..W$MODUL..MW)    CL
  DEL          ($XW..W$MODUL..WB)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```


| | | |
|---------------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 145 |
| 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

147

3
1
8

3.1.8.3. SIPE: DESCRIPTION OF STEPS

SIPE: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

PRODUCTION TURNOVER: PACR22

.Permanent input files:
- 'Batch' PEI file
PAC7AB : Physical name = \$PXU..\$PRO.\$PFI.AB
- 'On-line' PEI file
PAC7AC : Physical name = \$PXU..\$PRO.\$PFI.AC
- Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
- Index file
PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
- Error message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Transaction file:
- User input
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:
- Transactions used to build data cards
for TRANSFER utilities
PAC7MT : Physical name = \$PXU..W\$MODUL..MT

.Output reports:
- Execution report
SYS007
- Batch-procedure authorization option
SYS005

VERSIONING UTILITIES

PEI: PRODUCTION ENVIRONMENT INTERFACE

SIPE: PRODUCTION TURNOVER SIMULATION

3

1

8

3.1.8.4. SIPE: EXECUTION JCL

```

===SEQ FOR PEI
===MOD SIPE
* $X JOB JNM=$PRFJ.SIPE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.SIPE          ***** $PROD $PREL *****
*      PEI OPTION : SIMULATION
/* *****
/* INPUT : REQUIRED  '*' LINE
/*          REQUIRED  'EE' LINE
/*          (ENVIRONMENT IDENTIFICATION)
/*          ONE 'EU' LINE PER ENTITY
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL$) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE RESULT FILE MT --- */
DEL          ($PXU..R$MODUL..MT) CL
DEF CL      (NAME ($PXU..R$MODUL..MT) -
             MODEL($MODEL$) -
             REC (500 500) -
             RECSZ(80 80) -
             RECFM(FB(80)) -
             ) DATA (NAME ($PXU..R$MODUL..MT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORDLIB
/*
/. JCLST03
* --- STEP 03 --- PACR22
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AB,'$PXU..$PRO.$PFI.AB',,VSAM
// DLBL PAC7AC,'$PXU..$PRO.$PFI.AC',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7MT,'$PXU..R$MODUL..MT',,VSAM
// EXEC PACR22,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

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.

| | | |
|---------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 151 |
| PAC/TRANSFER | | 3 |
| TRUP: TRANSFER-PARAMETER UPDATE | | 2 |
| | | 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.

| | | |
|---------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 152 |
| PAC/TRANSFER | | 3 |
| TRUP: TRANSFER-PARAMETER UPDATE | | 2 |
| | | 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.

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

| | | |
|---------------------------------|------|-----|
| | PAGE | 153 |
| VERSIONING UTILITIES | | 3 |
| PAC/TRANSFER | | 2 |
| TRUP: TRANSFER-PARAMETER UPDATE | | 1 |

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

PAC/TRANSFER

TRUP: TRANSFER-PARAMETER UPDATE

3

2

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 ! uuuuuuuu ! User code !
! 11 !   8 ! pppppppp ! Password  !
-----

```

. Session-selection line

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

```

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

```

VERSIONING UTILITIES

PAC/TRANSFER

TRUP: TRANSFER-PARAMETER UPDATE

3
2
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

PAC/TRANSFER

TRUP: TRANSFER-PARAMETER UPDATE

3

2

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
!-----

```

. 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
!-----

```

EXAMPLES:

EXAMPLE 1

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

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

EXAMPLE 2

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

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

EXAMPLE 3

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

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

EXAMPLE 4

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

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

EXAMPLE 5

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

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

3.2.1.3. TRUP: DESCRIPTION OF STEPS

TRUP: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

UPDATE OF THE SELECTION PARAMETERS: PTUG10

This step updates the selection-parameter file.

.Permanent input files:

-Data file

PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

-Index file

PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN

-Error-message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Transaction file

-User input

PAC7MA : Physical name = \$XW..W\$MODUL..INPUT

.Output file:

-List of Transfer Sets

PAC7ML : Physical name = \$XW..W\$MODUL..ML

.Input/output file:

-Parameter file

PAC7UV : Physical name = \$PXU..\$PRO.\$PFI.UV

.Work file:

-Transaction file with generated multiple deletions

PAC7MV : Physical name = \$XW..W\$MODUL..MV

.Output reports:

-Input check

SYS005

-User check

SYS007

SELECTION-PARAMETER PRINTOUT: PTUG11

.Permanent input files:
-Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Error-message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
-Parameter file
 PAC7UV : Physical name = \$PXU..\$PRO.\$PFI.UV

.Output file:
-List of target sessions
 PAC7GL : Physical name = \$XW..W\$MODUL..GL

.Output report:
-Printout of parameter table
 SYS005

PRINTING OF TARGET-SESSION LIST: PTUG12

.Input files:
-Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Parameter file
 PAC7UV : Physical name = \$PXU..\$PRO.\$PFI.UV
-Error-message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
-Target-session list
 PAC7GL : Physical name = \$XW..W\$MODUL..GL
-List of Sets
 PAC7ML : Physical name = \$XW..W\$MODUL..ML

.Output report:
-Target-session list printout
 SYS005

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRUP: TRANSFER-PARAMETER UPDATE

1

3.2.1.4. TRUP: EXECUTION JCL

```

===SEQ FOR PACTR
===MOD TRUP
* $X JOB JNM=$PRFJ.TRUP,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.TRUP          ***** $PROD $PREL *****
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
            MODEL($MODEL$) -
            REC (1 200) -
            ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE   GL --- */
DEL          ($XW..W$MODUL..GL) CL
DEF CL      (NAME ($XW..W$MODUL..GL) -
            MODEL($MODEL$) -
            REC (100 100) -
            RECSZ (19 19) -
            RECFM(FB(19)) -
            ) DATA (NAME ($XW..W$MODUL..GL.D) )
/* --- DELETE-DEFINE TEMPORARY FILE   ML --- */
DEL          ($XW..W$MODUL..ML) CL
DEF CL      (NAME ($XW..W$MODUL..ML) -
            MODEL($MODEL$) -
            REC (100 100) -
            RECSZ (80 80) -
            RECFM(FB(80)) -
            ) DATA (NAME ($XW..W$MODUL..ML.D) )
/* --- DELETE-DEFINE TEMPORARY FILE   MV --- */
DEL          ($XW..W$MODUL..MV) CL
DEF CL      (NAME ($XW..W$MODUL..MV) -
            MODEL($MODEL$) -
            REC (100 100) -
            RECSZ (80 80) -
            RECFM(FB(80)) -
            ) DATA (NAME ($XW..W$MODUL..MV.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
/*
/. JCLST03
* --- STEP 03 --- PTUG10
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MA,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7ML,'$XW..W$MODUL..ML',,VSAM
// DLBL PAC7MV,'$XW..W$MODUL..MV',,VSAM
// DLBL PAC7UV,'$PXU..$PRO.$PFI.UV',,VSAM
// ON $RC GE 8 GOTO STEPEND
// EXEC PTUG10,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTUG11

```


VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRUP: TRANSFER-PARAMETER UPDATE

1

```
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7GL,'$XW..W$MODUL..GL',,VSAM
// DLBL PAC7UV,'$PXU..$PRO.$PFI.UV',,VSAM
// EXEC PTUG11,SIZE=AUTO
/. JCLST05
* --- STEP 05 --- PTUG12
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7GL,'$XW..W$MODUL..GL',,VSAM
// DLBL PAC7ML,'$XW..W$MODUL..ML',,VSAM
// DLBL PAC7UV,'$PXU..$PRO.$PFI.UV',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTUG12,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
DEL ($XW..W$MODUL..GL) CL
DEL ($XW..W$MODUL..ML) CL
DEL ($XW..W$MODUL..MV) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```

| | | |
|---------------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 162 |
| PAC/TRANSFER | | 3 |
| 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.

RESULT

A smaller archived Journal including effective transactions only.

OUTPUT REPORT

Statistical data on the TRJC execution.

3.2.2.2. TRJC: USER INPUT

TRJC: USER INPUT

. User identification line (required)

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

. Options

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

VERSIONING UTILITIES
PAC/TRANSFER
TRJC: COMPRESSION OF ARCHIVED JOURNAL

PAGE

164

3
2
2

3.2.2.3. TRJC: DESCRIPTION OF STEPS

TRJC: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

COMPRESSION (FIRST STAGE): PTUG05

.Permanent input files:
-Sequential journal
 PAC7PJ : Physical name = \$PXU..SEQ.\$PRO\$PFI.PJ
 Tape is SYS015
-Index file
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
-Error-message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Transaction file:
-User input
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:
-Temporary journal
 PAC7GP : Physical name = \$XW..W\$MODUL..GP
 Tape is SYS012

.Output reports:
-Check on input: SYS005
-Batch procedure abend report SYS007

. Sort

COMPRESSION (SECOND STAGE): PTUG06

.Input transaction file:
-Temporary file
 PAC7GP : Physical name = \$XW..W\$MODUL..GP
 Tape is SYS011

.Output file:
-Sequential compressed file
 PAC7PK : Physical name = \$XW..W\$MODUL..PK
 Tape is SYS012

. Sort

CLASSIFICATION OF DELETIONS/CREATIONS: PTUG07

.Input file:
-Index file
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN

.Input transaction files:
-Temporary journal
 PAC7PK : Physical name = \$XW..W\$MODUL..PK
 Tape is SYS012

.Output file:
-Compressed sequential file
 PAC7PL : Physical name = \$PXU..SEQ.\$PRO\$PFI.JT
 Tape is SYS013

. Sort

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRJC: COMPRESSION OF ARCHIVED JOURNAL

2

3.2.2.4. TRJC: EXECUTION JCL

```

===SEQ FOR PACTR
===MOD TRJC
* $X JOB JNM=$PRFJ.TRJC,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.TRJC          ***** $PROD $PREL *****
*   COMPRESSION OF PACBASE ARCHIVED JOURNAL FILE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE GP --- */
DEL          ($XW..W$MODUL..GP) CL
DEF CL       (NAME ($XW..W$MODUL..GP) -
              MODEL($MODEL) -
              REC (1000 1000) -
              RECSZ(221 221) -
              RECFM(FB(221)) -
              ) DATA (NAME ($XW..W$MODUL..GP.D) )
/* --- DELETE-DEFINE FICHER SAUVEGARDE JT --- */
DEL          ($PXU..$PRO.$PFI..JT) CL
DEF CL       (NAME ($PXU..$PRO.$PFI..JT) -
              MODEL($MODEL) -
              REC (5000 1000) -
              RECSZ(165 165) -
              RECFM(FB(165)) -
              ) DATA (NAME ($PXU..$PRO.$PFI..JT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE PK --- */
DEL          ($XW..W$MODUL..PK) CL
DEF CL       (NAME ($XW..W$MODUL..PK) -
              MODEL($MODEL) -
              REC (5000 1000) -
              RECSZ(165 165) -
              RECFM(FB(165)) -
              ) DATA (NAME ($XW..W$MODUL..PK.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PTUG05
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7GP,'$XW..W$MODUL..GP',,VSAM
/* PAC7GP ON TAPE : SYS012 ASSIGNED
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7PJ,'$PXU..SEQ.$PRO$PFI..PJ',,VSAM
/* PAC7PJ ON TAPE : SYS015 ASSIGNED
// DLBL PAC7UV,'$PXU..$PRO.$PFI.UV',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 0 GOTO STEPEND
// EXEC PTUG05,SIZE=(AUTO,50K)

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRJC: COMPRESSION OF ARCHIVED JOURNAL

2

```

/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTUG06
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7GP,'$XW..W$MODUL..GP',,VSAM
/* PAC7GP ON TAPE : SYS011 ASSIGNED
// DLBL PAC7PK,'$XW..W$MODUL..PK',,VSAM
/* PAC7PK ON TAPE : SYS012 ASSIGNED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTUG06,SIZE=(AUTO,50K)
/. JCLST05
* --- STEP 05 --- PTUG07
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7PK,'$XW..W$MODUL..PK',,VSAM
/* PAC7PK ON TAPE : SYS012 ASSIGNED
// DLBL PAC7PL,'$PXU..$PRO.$PFI.JT',,VSAM
/* PAC7PL ON TAPE : SYS013 ASSIGNED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTUG07,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
DEL          ($XW..W$MODUL..GP)    CL
DEL          ($XW..W$MODUL..PK)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

| | | |
|------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 167 |
| PAC/TRANSFER | | 3 |
| TRPF: TRANSFER-FILE CREATION | | 2 |
| | | 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.

3.2.3.2. TRPF: USER INPUT

TRPF: USER INPUT

. User identification line (required)

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

. Transaction Set for processing selection line (required)

| !Pos.! | Len.! | Value | ! Meaning | ! |
|--------|-------|---------|--|---|
| ! 2 ! | 2 ! | 'LT' | ! | ! |
| ! 4 ! | 5 ! | 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).

3.2.3.3. TRPF: DESCRIPTION OF STEPS

TRPF: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

CREATION OF TRANSFER FILE: PTUG50

.Permanent input files:
-Index file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Error-message file
 PAC7AE : Physical name = \$PXU..\$PRO.\$PRO.AE
-Parameter file
 PAC7UV : Physical name = \$PXU..\$PRO.\$PFI.UV
-Sequential or compressed file
 PAC7JT : Physical name = \$PXU..SEQ.\$PRO\$PFI.JT
 Tape is SYS014

.Transaction file:
-User input
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output files:
-Sequential transfer journal
 PAC7TJ : Physical name = \$PXU..SEQ.\$PRO\$PFI.TJ
 Tape is SYS011

. Sort

.Output reports:
-Transfer statistics SYS005
-Check on user SYS007
-TRPF-transaction list SYS003

VERSIONING UTILITIES
 PAC/TRANSFER
 TRPF: TRANSFER-FILE CREATION

3
 2
 3

3.2.3.4. TRPF: EXECUTION JCL

```

===SEQ FOR PACTR
===MOD TRPF
* $X JOB JNM=$PRFJ.TRPF,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL,$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2,$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST3,$PWL
/* *****
// JOB $PRFJ.TRPF          ***** $PROD $PREL *****
*   CREATING OF SESSION TRANSFER FILE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND PRINTER L.U.
// ASSGN SYS003,$LST3 3TRD PRINTER L.U.
/* GOTO JCLSTXX
. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE FICHER SAUVEGARDE TJ --- */
DEL          ($PXU..$PRO.$PFI..TJ) CL
DEF CL       (NAME ($PXU..$PRO.$PFI..TJ) -
              MODEL($MODEL) -
              REC (5000 1000) -
              RECSZ(170 170) -
              RECFM(FB(170)) -
              ) DATA (NAME ($PXU..$PRO.$PFI..TJ.D) )
/*
. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
LTLLLLL
/*
. JCLST03
* --- STEP 03 --- PTUG50
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7JT,'$PXU..$PRO.$PFI.JT',,VSAM
/* PAC7JT ON TAPE : SYS014 ASSIGNED
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TJ,'$PXU..$PRO.$PFI.TJ',,VSAM
/* PAC7TJ ON TAPE : SYS011 ASSIGNED
// DLBL PAC7UV,'$PXU..$PRO.$PFI.UV',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTUG50,SIZE=(AUTO,50K)
. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

| | | |
|------------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 171 |
| PAC/TRANSFER | | 3 |
| TRDU: DSMS-ENVIRONMENT PREPARATION | | 2 |
| | | 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.

This update must be executed BEFORE the transfer operation.

| | | |
|------------------------------------|------|-----|
| | PAGE | 172 |
| 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.

3.2.4.2. TRDU: USER INPUT

TRDU: USER INPUT

. User identification line (required)

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

. TRANSACTION SET selection line (required)

| !Pos.! | Len.! | Value | ! Meaning | ! |
|--------|-------|---------|---------------------------------|---|
| ! 2 ! | 2 ! | 'LT' | ! | ! |
| ! 4 ! | 5 ! | 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

3

2

4

-Preparation file for target-Change authorizations
PAC7GC : Physical name = \$XW..W\$MODUL..GC

.Output files:

-Target before-transfer creation transactions
PAC7CC : Physical name = \$XW..W\$MODUL..CC
-Target after-transfer deletion transactions
PAC7SC : Physical name = \$XW..W\$MODUL..SC

. Sort

.Output report:

-Execution report

SYS005

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRDU: DSMS-ENVIRONMENT PREPARATION

4

3.2.4.4. TRDU: EXECUTION JCL

```

===SEQ FOR DSM
===SEQ FOR DSM
===MOD TRDU
* $X JOB JNM=$PRFJ.TRDU,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL,$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2,$PWL
/* *****
// JOB $PRFJ.TRDU          ***** $PROD $PREL *****
*   PREPARING THE DSMS ENVIRONMENT
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE RESULT FILE AP --- */
DEL          ($PXU..R$MODUL..AP) CL
DEF CL      (NAME ($PXU..R$MODUL..AP) -
             MODEL($MODEL) -
             REC (100 100) -
             RECSZ(250 250) -
             RECFM(FB(250)) -
             ) DATA (NAME ($PXU..R$MODUL..AP.D) )
/* --- DELETE-DEFINE RESULT FILE AV --- */
DEL          ($PXU..R$MODUL..AV) CL
DEF CL      (NAME ($PXU..R$MODUL..AV) -
             MODEL($MODEL) -
             REC (100 100) -
             RECSZ(250 250) -
             RECFM(FB(250)) -
             ) DATA (NAME ($PXU..R$MODUL..AV.D) )
/* --- DELETE-DEFINE TEMPORARY FILE BM --- */
DEL          ($XW..W$MODUL..BM) CL
DEF CL      (NAME ($XW..W$MODUL..BM) -
             MODEL($MODEL) -
             REC (1 1) -
             RECSZ(80 80) -
             RECFM(FB(80)) -
             ) DATA (NAME ($XW..W$MODUL..BM.D) )
/* --- DELETE-DEFINE TEMPORARY FILE CC --- */
DEL          ($XW..W$MODUL..CC) CL
DEF CL      (NAME ($XW..W$MODUL..CC) -
             MODEL($MODEL) -
             REC (100 100) -
             RECSZ(250 250) -
             RECFM(FB(250)) -
             ) DATA (NAME ($XW..W$MODUL..CC.D) )
/* --- DELETE-DEFINE TEMPORARY FILE CI --- */
DEL          ($XW..W$MODUL..CI) CL
DEF CL      (NAME ($XW..W$MODUL..CI) -
             MODEL($MODEL) -
             REC (100 100) -
             RECSZ(250 250) -
             RECFM(FB(250)) -
             ) DATA (NAME ($XW..W$MODUL..CI.D) )
/* --- DELETE-DEFINE TEMPORARY FILE GC --- */
DEL          ($XW..W$MODUL..GC) CL
DEF CL      (NAME ($XW..W$MODUL..GC) -
             MODEL($MODEL) -
             REC (100 100) -

```


VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRDU: DSMS-ENVIRONMENT PREPARATION

4

```

                RECSZ(40 40) -
                RECFM(FB(40)) -
                ) DATA (NAME ($XW..W$MODUL..GC.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      SC --- */
DEL             ($XW..W$MODUL..SC) CL
DEF CL         (NAME ($XW..W$MODUL..SC) -
                MODEL($MODEL) -
                REC (100 100) -
                RECSZ(250 250) -
                RECFM(FB(250)) -
                ) DATA (NAME ($XW..W$MODUL..SC.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      SI --- */
DEL             ($XW..W$MODUL..SI) CL
DEF CL         (NAME ($XW..W$MODUL..SI) -
                MODEL($MODEL) -
                REC (100 100) -
                RECSZ(250 250) -
                RECFM(FB(250)) -
                ) DATA (NAME ($XW..W$MODUL..SI.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
LTLLLLL
/*
/. JCLST03
* --- STEP 03 --- PTUG42
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7BM,'$XW..W$MODUL..BM',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7UV,'$PXU..$PRO.$PFI.UV',,VSAM
// EXEC PTUG42,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTUG44
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7CI,'$XW..W$MODUL..CI',,VSAM
// DLBL PAC7DC,'$DSMS.DC',,VSAM,CAT=DSVCT
// DLBL PAC7GC,'$XW..W$MODUL..GC',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..BM',,VSAM
// DLBL PAC7SI,'$XW..W$MODUL..SI',,VSAM
// DLBL PAC7UV,'$PXU..$PRO.$PFI.UV',,VSAM
// EXEC PTUG44,SIZE=AUTO
/. JCLST05
* --- STEP 05 --- PTUG46
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7CC,'$XW..W$MODUL..CC',,VSAM
// DLBL PAC7GC,'$XW..W$MODUL..GC',,VSAM
// DLBL PAC7SC,'$XW..W$MODUL..SC',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTUG46,SIZE=(AUTO,50K)
/. JCLST06
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AP,'$PXU..R$MODUL..AP',,VSAM,DISP=OLD
// DLBL PAC7AV,'$PXU..R$MODUL..AV',,VSAM,DISP=OLD
// DLBL PAC7CC,'$XW..W$MODUL..CC',,VSAM
// DLBL PAC7CI,'$XW..W$MODUL..CI',,VSAM
// DLBL PAC7SC,'$XW..W$MODUL..SC',,VSAM
// DLBL PAC7SI,'$XW..W$MODUL..SI',,VSAM
// EXEC IDCAMS,SIZE=AUTO
REPRO IFILE(PAC7SI) OFILE(PAC7AV)
REPRO IFILE(PAC7CC) OFILE(PAC7AV)
REPRO IFILE(PAC7SC) OFILE(PAC7AP)
REPRO IFILE(PAC7CI) OFILE(PAC7AP)

```

VERSIONING UTILITIES

PAC/TRANSFER

TRDU: DSMS-ENVIRONMENT PREPARATION

3

2

4

```
/*
/. STEPEND
* --- STEP 07 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
DEL          ($XW..W$MODUL..BM)   CL
DEL          ($XW..W$MODUL..CC)   CL
DEL          ($XW..W$MODUL..CI)   CL
DEL          ($XW..W$MODUL..GC)   CL
DEL          ($XW..W$MODUL..SC)   CL
DEL          ($XW..W$MODUL..SI)   CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

| | | |
|--|------|-----|
| | PAGE | 179 |
| 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 | 180 |
| 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 | 181 |
| 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).

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.

VERSIONING UTILITIES

PAC/TRANSFER

TRRP: GENERATION OF TRANSFER TRANSACTIONS

3

2

6

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 !
-----

```

3.2.6.3. TRRP: DESCRIPTION OF STEPS

TRRP: DESCRIPTION OF STEPS

PREPARATION OF EXTRACTION: PTUG60

.Permanent input files:
-Index file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Error messages
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
-Parameter-setting file
 PAC7UV : Physical name = \$PXU..\$PRO.\$PFI.UV
-Compressed journal file
 PAC7JT : Physical name = \$PXU..SEQ.\$PRO\$PFI.TJ
 Tape is SYS012

.Transaction file:
-User input
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:
-Parameter-line file
 PAC7BM : Physical name = \$XW..W\$MODUL..BM
-Temporary journal file
 PAC7PJ : Physical name = \$XW..W\$MODUL..PJ
 Tape is SYS015

.Output reports:
-Transfer statistics SYS005
-User check SYS007

.Return code(s):
 . 04 : There is not session to extract
EXTRACTION: PACX

This step extracts transactions based on user input.

.Permanent input files:
-Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Index file
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN
-Error-message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
-Transactions selected on Journal
 PAC7PJ : Physical name = \$XW..W\$MODUL..PJ

.Input transaction file:
-User input
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Work files
-User input
 PAC7BM : Physical name = \$XW..W\$MODUL..BM
-Journal transactions (EXPJ)
 PAC7MJ : Physical name = \$XW..W\$MODUL..MJ
-Extracted transactions
 PAC7WD : Physical name = \$XW..W\$MODUL..WD

.Output file:
-Transactions extracted for UPDT
 PAC7MV : Physical name = \$XW..W\$MODUL..MV

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRRP: GENERATION OF TRANSFER TRANSACTIONS

6

. Sort

.Output reports:

-General program-stream printout
 SYS005
-List of errors on input transactions
 SYS007
-Extraction list report(s)
 SYS003

.Return code(s):

. 0 : OK
. 08 : Fatal error ,it is in SYS007

POSITIONNING THE 'PROCESSED SESSION' ATTRIBUTE: PTUG61

.Permanent input files:

-Index file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Error-message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input transaction file

-User input
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Input/Output file:

-Parameter-settings
 PAC7UV : Physical name = \$PXU..\$PRO.\$PFI.UV

.Output report(s):

-Transfer statistics
 SYS005

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRRP: GENERATION OF TRANSFER TRANSACTIONS

6

3.2.6.4. TRRP: EXECUTION JCL

```

===MOD TRRP
* $X JOB JNM=$PRFJ.TRRP,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.TRRP          ***** $PROD $PREL *****
*   EXTRACTION FROM COMPRESSED JOURNAL FILE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL$) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE   BM --- */
DEL          ($XW..W$MODUL..BM) CL
DEF CL       (NAME ($XW..W$MODUL..BM) -
              MODEL($MODEL$) -
              REC (1 1) -
              RECSZ(80 80) -
              RECFM(FB(80)) -
              ) DATA (NAME ($XW..W$MODUL..BM.D) )
/* --- DELETE-DEFINE TEMPORARY FILE   ML --- */
DEL          ($XW..W$MODUL..ML) CL
DEF CL       (NAME ($XW..W$MODUL..ML) -
              MODEL($MODEL$) -
              REC (5000 1000) -
              RECSZ(127 127) -
              RECFM(FB(127)) -
              ) DATA (NAME ($XW..W$MODUL..ML.D) )
/* --- DELETE-DEFINE RESULT FILE     MV --- */
DEL          ($PXU..R$MODUL..MV) CL
DEF CL       (NAME ($PXU..R$MODUL..MV) -
              MODEL($MODEL$) -
              REC (5000 1000) -
              RECSZ(80 80) -
              RECFM(FB(80)) -
              ) DATA (NAME ($PXU..R$MODUL..MV.D) )
/* --- DELETE-DEFINE TEMPORARY FILE   PJ --- */
DEL          ($XW..W$MODUL..PJ) CL
DEF CL       (NAME ($XW..W$MODUL..PJ) -
              MODEL($MODEL$) -
              REC (5000 1000) -
              RECSZ(165 165) -
              RECFM(FB(165)) -
              ) DATA (NAME ($XW..W$MODUL..PJ.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
/*
/. JCLST03
* --- STEP 03 --- PTUG60
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7BM,'$XW..W$MODUL..BM',,VSAM
// DLBL PAC7JT,'$PXU..$PRO.$PFI.TJ',,VSAM
/* PAC7JT ON TAPE : SYS012 ASSIGNED

```

VERSIONING UTILITIES

3

PAC/TRANSFER

2

TRRP: GENERATION OF TRANSFER TRANSACTIONS

6

```

// DLBL PAC7MB, '$XW..W$MODUL.INPUT', ,VSAM
// DLBL PAC7PJ, '$XW..W$MODUL.PJ', ,VSAM
/* PAC7PJ ON TAPE : SYS015 ASSIGNED
// DLBL PAC7UV, '$PXU..$PRO.$PFI.UV', ,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTUG60,SIZE=AUTO
*USER      PASSWORD          LLLLL
J
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTU600
// DLBL IJSYSUC, '$VCAP', ,VSAM
// DLBL PAC7AE, '$PXY..$PRO.$PRO.AE', ,VSAM
// DLBL PAC7MB, '$XW..W$MODUL.BM', ,VSAM
// DLBL PAC7ML, '$XW..W$MODUL.ML', ,VSAM
// DLBL PAC7PJ, '$XW..W$MODUL.PJ', ,VSAM
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU600,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST05
* --- STEP 05 --- PTU610
// DLBL IJSYSUC, '$VCAP', ,VSAM
// DLBL PAC7AE, '$PXY..$PRO.$PRO.AE', ,VSAM
// DLBL PAC7ML, '$XW..W$MODUL.ML', ,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTU610,SIZE=(AUTO,50K)
/. JCLST06
* --- STEP 06 --- PTUG61
// DLBL IJSYSUC, '$VCAP', ,VSAM
// DLBL PAC7AE, '$PXY..$PRO.$PRO.AE', ,VSAM
// DLBL PAC7AR, '$PXU..$PRO.$PFI.AR', ,VSAM
// DLBL PAC7MB, '$XW..W$MODUL.BM', ,VSAM
// DLBL PAC7UV, '$PXU..$PRO.$PFI.UV', ,VSAM
// EXEC PTUG61,SIZE=AUTO
/. STEPEND
* --- STEP 07 --- IDCAMS
// DLBL IJSYSUC, '$VCAP', ,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
DEL          ($XW..W$MODUL..BM)    CL
DEL          ($XW..W$MODUL..ML)    CL
DEL          ($XW..W$MODUL..PJ)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|--|------|-----|
| | PAGE | 188 |
| 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 | 189 |
| 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 | 191 |
| 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

3

TEAMCONNECTION

3

TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION

1

3.3.1.3. TCGP: EXECUTION JCL

```

===SEQ FOR TEAM
===MOD TCGP
* $X JOB JNM=$PRFJ.TCGP,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.TCGP          ***** $PROD $PREL *****
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE TEMPORARY FILE      BJ --- */
DEL          ($XW..W$MODUL..BJ) CL
DEF CL      (NAME ($XW..W$MODUL..BJ) -
             MODEL($MODEL' ) -
             REC ( ) -
             RECSZ ( ) -
             RECFM(FB()) -
             ) DATA (NAME ($XW..W$MODUL..BJ.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      JB --- */
DEL          ($XW..W$MODUL..JB) CL
DEF CL      (NAME ($XW..W$MODUL..JB) -
             MODEL($MODEL' ) -
             REC ( ) -
             RECSZ ( ) -
             RECFM(FB()) -
             ) DATA (NAME ($XW..W$MODUL..JB.D) )
/*
/. JCLST02
* --- STEP 02 --- PTC100
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7BJ,'$XW..W$MODUL..BJ',,VSAM
// DLBL PAC7JB,'$XW..W$MODUL..JB',,VSAM
// DLBL SYSPAF,'$XW..SYSPAF',,VSAM
// DLBL PAC7TS,'$PXY..$PRO.$PRO.TS',,VSAM
// EXEC PTC100,SIZE=AUTO
/. STEPEND
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..BJ) CL
DEL          ($XW..W$MODUL..JB) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

| | | |
|---|------|-----|
| VERSIONING UTILITIES | PAGE | 194 |
| TEAMCONNECTION | | 3 |
| TCCI: INTER-ENVIRONMENT INTEGRITY CHECK | | 3 |
| | | 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

3

TEAMCONNECTION

3

TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

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 : Physical name = $PXY..$PRO.$PRO.TS
-Error messages
  PAC7AE : Physical name = $PXY..$PRO.$PRO.AE
-Data file
  PAC7AR : Physical name = $PXU..$PRO.$PFI.AR
-Index file
  PAC7AN : Physical name = $PXU..$PRO.$PFI.AN

.Input file:
-Manager's identification
  PAC7CA : Physical name = $XW..W$MODUL..CA

.Work file:
-PAF standard KSDS file
  SYSPAF : Physical name = $XW..W$MODUL..SYSPAF

.Output report:
-Detected errors
          SYS005

.Output file:
-Extracted transactions
  PAC7RT : Physical name = $XW..W$MODUL..RT
```

INTER-ENVIRONMENT CHECK: PTC440

```
.Permanent input files:
-LIBRARY and SESSION file
  PAC7TS : Physical name = $PXY..$PRO.$PRO.TS
-Error messages
  PAC7AE : Physical name = $PXY..$PRO.$PRO.AE

.Input files:
-Manager's identification
  PAC7CA : Physical name = $XW..W$MODUL..CA
-List of generated objects found in TeamConnection
  PAC7UN : Physical name = $XW..W$MODUL..UN
-List of generated objects stored in VA Pac
  PAC7UM : Physical name = $XW..W$MODUL..UM

.Output report:
-Detected errors
          SYS005

.Output file:
-'Correction'-transaction file
  PAC7UR : Physical name = $XW..W$MODUL..UR
```

GENERATION OF BATCH UPDATE TRANSACTIONS: PTC220

```
.Permanent input files:
-Error messages
  PAC7AE : Physical name = $PXY..$PRO.$PRO.AE
-Data file
  PAC7AR : Physical name = $PXU..$PRO.$PFI.AR
-Index file
  PAC7AN : Physical name = $PXU..$PRO.$PFI.AN

.Input file:
-'Correction' transactions
  PAC7UR : Physical name = $XW..W$MODUL..UR
```

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

2

.Work file:

-PAF standard KSDS file

SYSPAF : Physical name = \$XW..W\$MODUL..SYSPAF

.Output report:

-Check report

SYS005

.Output file:

-VA-Pachbase update transactions

PAC7MV : Physical name = \$XW..W\$MODUL..MV

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

2

3.3.2.3. TCCI: EXECUTION

```

===SEQ FOR TEAM
===MOD TCCI
* $X JOB JNM=$PRFJ.TCCI,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.TCCI          ***** $PROD $PREL *****
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE TEMPORARY FILE      CA --- */
DEL          ($XW..W$MODUL..CA) CL
DEF CL      (NAME ($XW..W$MODUL..CA) -
            MODEL($MODEL$) -
            REC      () -
            RECSZ   () -
            RECFM(FB()) -
            ) DATA (NAME ($XW..W$MODUL..CA.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      MV --- */
DEL          ($XW..W$MODUL..MV) CL
DEF CL      (NAME ($XW..W$MODUL..MV) -
            MODEL($MODEL$) -
            REC      () -
            RECSZ   () -
            RECFM(FB()) -
            ) DATA (NAME ($XW..W$MODUL..MV.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      RT --- */
DEL          ($XW..W$MODUL..RT) CL
DEF CL      (NAME ($XW..W$MODUL..RT) -
            MODEL($MODEL$) -
            REC      () -
            RECSZ   () -
            RECFM(FB()) -
            ) DATA (NAME ($XW..W$MODUL..RT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      UM --- */
DEL          ($XW..W$MODUL..UM) CL
DEF CL      (NAME ($XW..W$MODUL..UM) -
            MODEL($MODEL$) -
            REC      () -
            RECSZ   () -
            RECFM(FB()) -
            ) DATA (NAME ($XW..W$MODUL..UM.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      UN --- */
DEL          ($XW..W$MODUL..UN) CL
DEF CL      (NAME ($XW..W$MODUL..UN) -
            MODEL($MODEL$) -
            REC      () -
            RECSZ   () -
            RECFM(FB()) -
            ) DATA (NAME ($XW..W$MODUL..UN.D) )
/* --- DELETE-DEFINE TEMPORARY FILE      UR --- */
DEL          ($XW..W$MODUL..UR) CL
DEF CL      (NAME ($XW..W$MODUL..UR) -
            MODEL($MODEL$) -
            REC      () -
            RECSZ   () -
            RECFM(FB()) -
            ) DATA (NAME ($XW..W$MODUL..UR.D) )
/*
/. JCLST02
* --- STEP 02 --- PTC400
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM

```

VERSIONING UTILITIES

3

TEAMCONNECTION

3

TCCI: INTER-ENVIRONMENT INTEGRITY CHECK

2

```

// DLBL PAC7CA, '$XW.W$MODUL.CA', ,VSAM
// DLBL PAC7RT, '$XW.W$MODUL.RT', ,VSAM
// DLBL SYSPAF, '$XW.SYSPAF', ,VSAM
// DLBL PAC7TS, '$PXU..$PRO.$PRO.TS', ,VSAM
// EXEC PTC400,SIZE=AUTO
/. JCLST03
* --- STEP 03 --- PTC440
// DLBL IJSYSUC, '$VCAP', ,VSAM
// DLBL PAC7AE, '$PXU..$PRO.$PRO.AE', ,VSAM
// DLBL PAC7CA, '$XW.W$MODUL.CA', ,VSAM
// DLBL PAC7TS, '$PXU..$PRO.$PRO.TS', ,VSAM
// DLBL PAC7UM, '$XW.W$MODUL.UM', ,VSAM
// DLBL PAC7UN, '$XW.W$MODUL.UN', ,VSAM
// DLBL PAC7UR, '$XW.W$MODUL.UR', ,VSAM
// EXEC PTC440,SIZE=AUTO
/. JCLST04
* --- STEP 04 --- PTC220
// DLBL IJSYSUC, '$VCAP', ,VSAM
// DLBL PAC7AE, '$PXU..$PRO.$PRO.AE', ,VSAM
// DLBL PAC7AN, '$PXU..$PRO.$PFI.AN', ,VSAM
// DLBL PAC7AR, '$PXU..$PRO.$PFI.AR', ,VSAM
// DLBL PAC7MV, '$XW.W$MODUL.MV', ,VSAM
// DLBL SYSPAF, '$XW.SYSPAF', ,VSAM
// DLBL PAC7UR, '$XW.W$MODUL.UR', ,VSAM
// EXEC PTC220,SIZE=AUTO
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC, '$VCAP', ,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW.W$MODUL.CA)    CL
DEL          ($XW.W$MODUL.MV)    CL
DEL          ($XW.W$MODUL.RT)    CL
DEL          ($XW.W$MODUL.UM)    CL
DEL          ($XW.W$MODUL.UN)    CL
DEL          ($XW.W$MODUL.UR)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

| | | |
|------------------------------|------|-----|
| VERSIONING UTILITIES | PAGE | 199 |
| 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.

3.3.3.2. TCLS: USER INPUT

TCLS : ENTREES UTILISATEUR

USER INPUT

One line for each update request.

Parameter line for target session update:

```
-----  
!Pos.! Lon.! Value  ! Meaning                                     !  
!-----+-----+-----+-----!  
! 1 ! 1 ! 'C'      ! Transaction code                             !  
!   !   ! 'M'      ! 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
 TEAMCONNECTION
 TCLS: LIBRARY-SESSION UPDATE

3
 3
 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
TEAMCONNECTION
TCLS: LIBRARY-SESSION UPDATE

PAGE

202

3
3
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 : Physical name = \$PXY..\$PRO.\$PRO.TS

.Permanent input file:
-Error messages
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input transaction file:
-User input
PAC7MV : Physical name = \$XW..W\$MODUL..MV

.Output report:
-Update report
SYS005

PRINTING OF LIBRARY AND SESSION FILE: PTC030

.Permanent input files:
-TARGET LIBRARIES and SESSIONS
PAC7TS : Physical name = \$PXY..\$PRO.\$PRO.TS
-Error messages
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Output report:
-List of TARGET LIBRARY and SESSIONS
SYS005

VERSIONING UTILITIES
 TEAMCONNECTION
 TCLS: LIBRARY-SESSION UPDATE

3
 3
 3

3.3.3.4. TCLS: EXECUTION JCL

```

===SEQ FOR TEAM
===MOD TCLS
* $X JOB JNM=$PRFJ.TCLS,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.TCLS          ***** $PROD $PREL *****
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE TEMPORARY FILE      MV --- */
DEL          ($XW..W$MODUL..MV) CL
DEF CL      (NAME ($XW..W$MODUL..MV) -
             MODEL($MODEL' ) -
             REC ( ) -
             RECSZ ( ) -
             RECFM(FB( ) ) -
             ) DATA (NAME ($XW..W$MODUL..MV.D) )
/*
/. JCLST02
* --- STEP 02 --- PTC010
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MV,'$XW..W$MODUL..MV',,VSAM
// DLBL PAC7TS,'$PXY..$PRO.$PRO.TS',,VSAM
// EXEC PTC010,SIZE=AUTO
/. JCLST03
* --- STEP 03 --- PTC030
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7TS,'$PXY..$PRO.$PRO.TS',,VSAM
// EXEC PTC030,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..MV)    CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

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

PAGE 204

4

4. MANAGER'S UTILITIES

| | | |
|---|------|-----|
| MANAGER'S UTILITIES | PAGE | 205 |
| STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES | | 4 |
| STOP: INTRODUCTION | | 1 |
| | | 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

STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES

STOP: USER INPUT

4

1

2

4.1.2. STOP: USER INPUT

STOP: USER INPUT

.One command line:

```

-----
! POS.! LEN.! VALUE ! MEANING !
!-----!
! 2 ! 1 ! ' ' ! Blank line code !
! 3 ! 8 !nnnnnnn! 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

STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES

STOP: DESCRIPTION OF STEPS

4

1

3

4.1.3. STOP: DESCRIPTION OF STEPS

STOP: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001STORAGE OPTIMIZATION OF DATA: PTUR00

.Permanent input files:
-Sequential image of the database
 PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI.PC
If backup Dispatch option :
-Sequential image #2 of the database
 PAC7PD : Physical name = \$PXU..SEQ.\$PRO\$PFI.PD

.Input transaction file:
-User transaction
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:
-Sequential image of the database
 PAC7CP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPC
If Dispatch backup option :
-Sequential image #2 of the database
 PAC7DP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPD

. Sort

.Output report:
-Execution report

SYS005

MANAGER'S UTILITIES

STOP: STORAGE OPTIMIZATION OF MULTI-VOLUMES

STOP: EXECUTION JCL

4

1

4

4.1.4. STOP: EXECUTION JCL

```

===MOD STOP
* $X JOB JNM=$PRFJ.STOP,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.STOP          ***** $PROD $PREL *****
*           STORAGE OPTIMIZATION OF PACBASE DATA
/*           (FOR MULTI-VOLUMES AR FILE)
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL$) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
* $X SLI MEM=DF$PRO.$PFI.PC.$SLIT
* $X SLI MEM=DF$PRO.$PFI.PD.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
NNNNNNNNDDDDDDDBBBBBBB
/*
/. JCLST03
* --- STEP 03 --- PTUR00
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7CP,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7CP ON TAPE : SYS011 ASSIGNED
// DLBL PAC7DP,'$PXU..SEQ.$PRO$PFI..NEWPD',,VSAM
/* PAC7DP ON TAPE : SYS012 ASSIGNED
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..PC',,VSAM
/* PAC7PC ON TAPE : SYS014 ASSIGNED
// DLBL PAC7PD,'$PXU..SEQ.$PRO$PFI..PD',,VSAM
/* PAC7PD ON TAPE : SYS015 ASSIGNED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTUR00,SIZE=(AUTO,50K)
/. JCLST04
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PD.$SLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```


| | | |
|---------------------------|------|-----|
| | PAGE | 209 |
| 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 | 210 |
| 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
SESSION MANAGEMENT
ESES: EXTRACTION OF SESSION NUMBERS

PAGE

211

4
2
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: EXECUTION JCL

4
 2
 4

4.2.4. ESES: EXECUTION JCL

```

===MOD ESES
* $X JOB JNM=$PRFJ.ESES,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.ESES          ***** $PROD $PREL *****
*          SESSION NUMBER CORRESPONDENCE TABLE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  /* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
  DEL          ($XW..W$MODUL..INPUT) CL
  DEF CL       (NAME ($XW..W$MODUL..INPUT) -
               MODEL($MODEL) -
               REC (1 200) -
               ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
  /* --- DELETE-DEFINE RESULT FILE MV --- */
  DEL          ($PXU..R$MODUL..MV) CL
  DEF CL       (NAME ($PXU..R$MODUL..MV) -
               MODEL($MODEL) -
               REC (100 1) -
               RECSZ(5 5) -
               RECFM(FB(5)) -
               ) DATA (NAME ($PXU..R$MODUL..MV.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
  *USER      PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PTUESS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7MV,'$PXU..R$MODUL..MV',,VSAM
// DLBL PAC7PC,'$PXU..$PRO.$PFI.PC',,VSAM
// EXEC PTUESS,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
  DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|--------------------------------------|------|-----|
| | PAGE | 214 |
| 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 : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input file (from ESES procedure):
-Session-number correspondence table
 PAC7MV : Physical name = \$PXU..R\$MODUL..MV

.Transaction file:
-User input
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Retrieval of the VisualAge Pacbase database backup
-Input
 PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI.PC
 Tape is SYS024
 If Dispatch option of the backup:
 PAC7PD : Physical name = \$PXU..SEQ.\$PRO\$PFI.PD
 Tape is SYS025
-Output
 PAC7CP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPC
 Tape is SYS012
 If Dispatch option of the backup:
 PAC7DP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPD
 Tape is SYS016

.Retrieval of the VisualAge Pacbase archived journal:
-Input
 PAC7PJ : Physical name = \$PXU..SEQ.\$PRO\$PFI.PJ
 Tape is SYS027
-Output
 PAC7JP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPJ
 Tape is SYS021

.Retrieval of the VA Pac generation-print request backup:
-Input
 PAC7PG : Physical name = \$PXU..SEQ.\$PRO\$PFI.PG
 Tape is SYS026
-Output
 PAC7GP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPG
 Tape is SYS018

.Retrieval of the PEI backup:
-Input
 PAC7PP : Physical name = \$PXU..SEQ.\$PRO\$PFI.PP
 Tape is SYS028
-Output
 PAC7EP : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPP
 Tape is SYS017

If DSMS is installed:

.Retrieval of the DSMS database backup:
-Input

MANAGER'S UTILITIES
SESSION MANAGEMENT
CSES: DESCRIPTION OF STEPS

4
2
7

PACDBB : Physical name = \$DSMS.BB
-Output
PACDJB : Physical name = \$DSMS.NEWBB

.Retrieval of the DSMS archived journal:
-Input
PACDDJ : Physical name = \$DSMS.BJ
-Output
PAC7JD : Physical name = \$DSMS.NEWBJ

If Pactables is installed:

.Retrieval of the Pactables database backup:
-Input
PACDTC : Physical name = \$TXU..R\$MODUL..TC
-Output
PACDCT : Physical name = \$TXU..R\$MODUL..CT

.Output reports:
-Execution report
SYS005
-Batch-procedure authorization option
SYS007

MANAGER'S UTILITIES
SESSION MANAGEMENT
CSES: EXECUTION JCL

4
2
8

4.2.8. CSES: EXECUTION JCL

```

===MOD CSES
* $X JOB JNM=$PRFJ.CSES,CLASS=$PCJ,DISP=$PDJ,USER=' $PUJ '$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.CSES          ***** $PROD $PREL *****
*          COMPRESSION OF SESSION NUMBERS
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
* $X SLI MEM=DF$PRO.$PFI.PC.SSLIT
* $X SLI MEM=DF$PRO.$PFI.PD.SSLIT
/* --- DELETE-DEFINE BACKUP FILE          PG --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPG) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWPG) -
             MODEL($MODEL) -
             REC (100 100) -
             RECSZ(150 150) -
             RECFM(FB(150)) -
             ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPG.D) )
/* --- DELETE-DEFINE BACKUP FILE          PJ --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPJ) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ) -
             MODEL($MODEL) -
             REC ($RAJP $RAJS) -
             RECSZ(167 167) -
             RECFM(FB(167)) -
             ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPJ.D) )
/* --- DELETE-DEFINE BACKUP FILE          PP --- */
DEL          ($PXU..SEQ.$PRO$PFI..NEWPP) CL
DEF CL      (NAME ($PXU..SEQ.$PRO$PFI..NEWPP) -
             MODEL($MODEL) -
             REC (200 200) -
             RECSZ(110 110) -
             RECFM(FB(110)) -
             ) DATA (NAME ($PXU..SEQ.$PRO$PFI..NEWPP.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
SPC PG PJ PP
/*
/. JCLST03
* --- STEP 03 --- PTUCSS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
===SEQ FOR DSM
// DLBL PACDBB,'$DSMS.BB',,VSAM,CAT=DSVCT
// DLBL PACDJB,'$DSMS.NEWBB',,VSAM,CAT=DSVCT
// DLBL PACDDJ,'$DSMS.BJ',,VSAM,CAT=DSVCT
// DLBL PAC7JD,'$DSMS.NEWBJ',,VSAM,CAT=DSVCT
===SEQ

```

MANAGER'S UTILITIES
SESSION MANAGEMENT
CSES: EXECUTION JCL

4
2
8

```
// DLBL PAC7CP,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7CP ON TAPE : SYS012 ASSIGNED
// DLBL PAC7DP,'$PXU..SEQ.$PRO$PFI..NEWPD',,VSAM
/* PAC7DP ON TAPE : SYS016 ASSIGNED
// DLBL PAC7EP,'$PXU..SEQ.$PRO$PFI..NEWPP',,VSAM
/* PAC7EP ON TAPE : SYS017 ASSIGNED
// DLBL PAC7GP,'$PXU..SEQ.$PRO$PFI..NEWPG',,VSAM
/* PAC7GP ON TAPE : SYS018 ASSIGNED
// DLBL PAC7JP,'$PXU..SEQ.$PRO$PFI..NEWPJ',,VSAM
/* PAC7JP ON TAPE : SYS021 ASSIGNED
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7MV,'$PXU..RESES.MV',,VSAM
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..PC',,VSAM
/* PAC7PC ON TAPE : SYS024 ASSIGNED
// DLBL PAC7PD,'$PXU..SEQ.$PRO$PFI..PD',,VSAM
/* PAC7PD ON TAPE : SYS025 ASSIGNED
// DLBL PAC7PG,'$PXU..SEQ.$PRO$PFI..PG',,VSAM
/* PAC7PG ON TAPE : SYS026 ASSIGNED
// DLBL PAC7PJ,'$PXU..SEQ.$PRO$PFI..PJ',,VSAM
/* PAC7PJ ON TAPE : SYS027 ASSIGNED
// DLBL PAC7PP,'$PXU..SEQ.$PRO$PFI..PP',,VSAM
/* PAC7PP ON TAPE : SYS028 ASSIGNED
===SEQ FOR TAB
// DLBL PACDTC,'$TXU..R$MODUL..TC',,VSAM
// DLBL PACDCT,'$TXU..R$MODUL..NEWCT',,VSAM
===SEQ
// ON $RC GT 0 GOTO STEPEND
// EXEC PTUCSS,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PD.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PG.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PJ.$SLIT
* $X SLI MEM=RG$PRO.$PFI.PP.$SLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

| | PAGE | 220 |
|------------------------------------|------|-----|
| MANAGER'S UTILITIES | | 4 |
| GBIR: PARTITIONED DATABASE MANAGER | | 3 |
| GBIR: INTRODUCTION | | 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 | 221 |
| 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 | 222 |
|------------------------------------|------|-----|
| 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).

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

| |
|---|
| 4 |
| 3 |
| 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.

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

4
 3
 2

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 !uuuuuuu! 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

225

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 : Physical name = \$PXY..\$PRO.\$PRO.AE

.Transaction file:
-User input
PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Input files from EXSN:
-Master sub-network
PAC7MA : Physical name = \$PXU..R\$MODUL.BB
Tape is SYS013
-Slave sub-network
PAC7ES : Physical name = \$PXU..R\$MODUL.BB
Tape is SYS011

.Output file:
-Update transactions and sort criterion
PAC7MK : Physical name = \$XW..W\$MODUL..MK
Tape is SYS015

.Output reports:
-Report
SYS005
-Batch-procedure authorization option
SYS007

.Return codes:
. 0 : OK
. 08 : error or No authorization for batch procedure

SORT OF COMPARISON TRANSACTIONS: SORT

.Input file:
-Transactions with sort criteria
SORTIN : Physical name = \$XW..W\$MODUL..MK

.Output file:
-Sorted transactions
SORTOUT : Physical name = \$XW..W\$MODUL..KM

. Sort

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 : Physical name = \$PXY..\$PRO.\$PRO.AE

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

CPSN: SUB-NETWORK COMPARISON

4

3

2

.Input work file:

-Sorted generated transactions

PAC7MK : Physical name = \$XW..W\$MODUL..MK
Tape is SYS013

.Output file:

-Transactions generated for update

PAC7MB : Physical name = \$XW..W\$MODUL..MB
Tape is SYS012

.Output report:

-Generated transactions

SYS005

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

CPSN: SUB-NETWORK COMPARISON

4

3

2

4.3.2.4. CPSN: EXECUTION JCL

```

===MOD CPSN
* $X JOB JNM=$PRFJ.CPSN,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.CPSN          ***** $PROD $PREL *****
*       LIBRARY COMPARISON UTILITY
*       SUB-NETWORK COMPARISON
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE RESULT FILE MB --- */
DEL          ($PXU..R$MODUL..MB) CL
DEF CL       (NAME ($PXU..R$MODUL..MB) -
              MODEL($MODEL) -
              REC (8000 500) -
              RECSZ(80 80) -
              RECFM(FB(80)) -
              ) DATA (NAME ($PXU..R$MODUL..MB.D) )
/* --- DELETE-DEFINE TEMPORARY FILE MK --- */
DEL          ($XW..W$MODUL..MK) CL
DEF CL       (NAME ($XW..W$MODUL..MK) -
              MODEL($MODEL) -
              REC (15000 1000) -
              RECSZ(90 90) -
              RECFM(FB(90)) -
              ) DATA (NAME ($XW..W$MODUL..MK.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PTU850
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7ES,'$PXU..REXSNE.BB',,VSAM
/* PAC7ES ON TAPE : SYS011 ASSIGNED
// DLBL PAC7MA,'$PXU..REXSNM.BB',,VSAM
/* PAC7MA ON TAPE : SYS013 ASSIGNED
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7MK,'$XW..W$MODUL..MK',,VSAM
/* PAC7MK ON TAPE : SYS015 ASSIGNED
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU850,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/*
/. JCLST04
* --- STEP 04 --- SORT
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL SORTIN1,'$XW..W$MODUL..MK',,VSAM
// DLBL SORTOUT,'$XW..W$MODUL..MK',,VSAM,DISP=NEW
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION

```

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

CPSN: SUB-NETWORK COMPARISON

4
3
2

```
// EXEC SORT,SIZE=(AUTO,50K)
SORT FIELDS=(3,2,CH,A,6,7,CH,A)
RECORD TYPE=F,LENGTH=(90)
INPFIL VSAM
OUTFIL ESDS
/*
/. JCLST05
* --- STEP 05 --- PTU855
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7MB,'$PXU..R$MODUL..MB',,VSAM
/* PAC7MB ON TAPE : SYS012 ASSIGNED
// DLBL PAC7MK,'$XW..W$MODUL..MK',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7MK ON TAPE : SYS013 ASSIGNED
// EXEC PTU855,SIZE=AUTO
/. STEPEND
* --- STEP 06 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
DEL ($XW..W$MODUL..MK) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
```

| | | |
|------------------------------------|------|-----|
| MANAGER'S UTILITIES | PAGE | 229 |
| GBIR: PARTITIONED DATABASE MANAGER | | 4 |
| SASN: SUB-NETWORK BACKUP | | 3 |
| | | 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

GBIR: PARTITIONED DATABASE MANAGER

SASN: SUB-NETWORK BACKUP

4

3

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

4
 3
 3

4.3.3.3. SASN: DESCRIPTION OF STEPS

SASN: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

DATABASE VALIDATION: PTU130

This program is always executed.

.Permanent input files:
 -Error message file
 PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE
 -Data file
 PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
 -Index file
 PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN

 .Transaction input file:
 -Database-selection transactions
 PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

 .Output files:
 -Sequential data image:
 PAC7RP : Physical name = \$XW..W\$MODUL..RP
 Tape is SYS017
 (Must be able to contain all data)
 -Sequential index image
 PAC7NA : Physical name = \$XW..W\$MODUL..NA
 Tape is SYS016
 (Must be able to contain all indexes)
 - Image séquentielle données historisées
 PAC7RA : Physical name = \$XW..W\$MODUL..RA

 . Sort

 .Output reports:
 -Execution report
 SYS005
 -Batch-procedure authorization option
 SYS007

 .Return codes:
 . 0 : OK
 . 5 : Less than one library do not exist ,about those
 which are selected.
 . 6 : More than 99 movements (libraries) as input data
 . 8 : No autorization for batch procedure

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 : Physical name = \$PXY..\$PRO.\$PRO.AE

 .Input work files:
 -Data sequential image
 PAC7RP : Physical name = \$XW..W\$MODUL..RP
 Tape is SYS016
 -Index sequential image
 PAC7NA : Physical name = \$XW..W\$MODUL..NA
 Tape is SYS013
 -Frozen data sequential image
 PAC7RA : Physical name = \$XW..W\$MODUL..RA

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

SASN: SUB-NETWORK BACKUP

4

3

3

```
Tape is                SYS015

.Output file:
-Database sequential image
  PAC7SR : Physical name = $PXU..R$MODUL.SN
  Tape is                SYS014
If Dispatch option:
-Database sequential image 3/2
  PAC7PD : Physical name = $PXU..R$MODUL.SN2
  Tape is                SYS011

. Sort

.Output report:
-Execution report

                SYS005
```


MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

SASN: SUB-NETWORK BACKUP

4

3

3

4.3.3.4. SASN: EXECUTION JCL

```

===MOD SASN
* $X JOB JNM=$PRFJ.SASN, CLASS=$PCJ, DISP=$PDJ, USER=' $PUJ' $PWJ
* $X LST CLASS=$PCL, DISP=$PDL$PWL
* $X LST CLASS=$PCL, DISP=$PDL, LST=$LST2$PWL
/* *****
// JOB $PRFJ.SASN          ***** $PROD $PREL *****
*          LIBRARY COMPARISON UTILITY          *
*          SUB-NETWORK BACKUP                  *
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE NA --- */
DEL          ($XW..W$MODUL..NA) CL
DEF CL       (NAME ($XW..W$MODUL..NA) -
              MODEL($MODEL) -
              REC (1000 500) -
              RECSZ(55 55) -
              RECFM(FB(55)) -
              ) DATA (NAME ($XW..W$MODUL..NA.D) )
/* --- DELETE-DEFINE TEMPORARY FILE RA --- */
DEL          ($XW..W$MODUL..RA) CL
DEF CL       (NAME ($XW..W$MODUL..RA) -
              MODEL($MODEL) -
              REC (1000 500) -
              RECSZ(149 149) -
              RECFM(FB(149)) -
              ) DATA (NAME ($XW..W$MODUL..RA.D) )
/* --- DELETE-DEFINE TEMPORARY FILE RP --- */
DEL          ($XW..W$MODUL..RP) CL
DEF CL       (NAME ($XW..W$MODUL..RP) -
              MODEL($MODEL) -
              REC (1000 500) -
              RECSZ(149 149) -
              RECFM(FB(149)) -
              ) DATA (NAME ($XW..W$MODUL..RP.D) )
/* --- DELETE-DEFINE RESULT FILE SN --- */
DEL          ($PXU..R$MODUL..SN) CL
DEF CL       (NAME ($PXU..R$MODUL..SN) -
              MODEL($MODEL) -
              REC (1000 500) -
              RECSZ(63 159) -
              ) DATA (NAME ($PXU..R$MODUL..SN.D) )
/* --- DELETE-DEFINE RESULT FILE S2 --- */
DEL          ($PXU..R$MODUL..S2) CL
DEF CL       (NAME ($PXU..R$MODUL..S2) -
              MODEL($MODEL) -
              REC (1000 500) -
              RECSZ(63 159) -
              ) DATA (NAME ($PXU..R$MODUL..S2.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
QTF

```

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

SASN: SUB-NETWORK BACKUP

4

3

3

```

/*
/. JCLST03
* --- STEP 03 --- PTU130
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7NA,'$XW..W$MODUL..NA',,VSAM
/* PAC7NA ON TAPE : SYS014 ASSIGNED
// DLBL PAC7RA,'$XW..W$MODUL..RA',,VSAM
/* PAC7RA ON TAPE : SYS016 ASSIGNED
// DLBL PAC7RP,'$XW..W$MODUL..RP',,VSAM
/* PAC7RP ON TAPE : SYS017 ASSIGNED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// ON $RC GT 0 GOTO STEPEND
// EXEC PTU130,SIZE=(AUTO,50K)
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- PTU140
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7NA,'$XW..W$MODUL..NA',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7NA ON TAPE : SYS013 ASSIGNED
// DLBL PAC7PD,'$PXU..R$MODUL..SN2',,VSAM
/* PAC7PD ON TAPE : SYS011 ASSIGNED
// DLBL PAC7RA,'$XW..W$MODUL..RA',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7RA ON TAPE : SYS015 ASSIGNED
// DLBL PAC7RP,'$XW..W$MODUL..RP',,VSAM,DISP=(,DELETE,KEEP)
/* PAC7RP ON TAPE : SYS016 ASSIGNED
// DLBL PAC7SR,'$PXU..R$MODUL..SN',,VSAM
/* PAC7SR ON TAPE : SYS014 ASSIGNED
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTU140,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL ($XW..W$MODUL..INPUT) CL
DEL ($XW..W$MODUL..NA) CL
DEL ($XW..W$MODUL..RA) CL
DEL ($XW..W$MODUL..RP) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

EMSN: EXTRACTION FOR SUB-NETWORK MERGE

PAGE

235

4

3

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

GBIR: PARTITIONED DATABASE MANAGER

EMSN: EXTRACTION FOR SUB-NETWORK MERGE

4

3

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

4

3

4

4.3.4.3. EMSN: DESCRIPTION OF STEPS

EMSN: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

SUB-NETWORK EXTRACTION: PTU810

This step may extract up to 99 libraries.

.Permanent input files:

-Index file

PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN

-Data file

PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

-Error message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Transaction file:

-User input

PAC7ME : Physical name = \$XW..W\$MODUL..INPUT

.Output file:

-Extracted sub-network

PAC7BB : Physical name = \$PXU..R\$MODUL..BB

.Output reports:

-Lines required as MESN input

SYS007

-Extraction report

SYS003

-Batch-procedure authorization option

SYS005

.Sort

.Return codes:

. 0 : OK

. 8 : error or No authorization for batch procedure

The return code is set when the EMSN procedure is immediately followed by the execution of the MESN procedure.

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

EMSN: EXTRACTION FOR SUB-NETWORK MERGE

4

3

4

4.3.4.4. EMSN: EXECUTION JCL

```

===MOD EMSN
* $X JOB JNM=$PRFJ.EMSN,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL,$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST3$PWL
/* *****
// JOB $PRFJ.EMSN          ***** $PROD $PREL *****
*      LIBRARY COMPARISON UTILITY
*              SUB-NETWORK EXTRACTION FOR MERGE
/* *****
/* INPUT  : ONE USER AND LIBRARY LINE FOR EACH LIBRARY OF THE
/*              SUB-NETWORK TO BE EXTRACTED (99 LINES MAXIMUM)
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
// ASSGN SYS003,$LST3 3TRD PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE RESULT FILE BB --- */
DEL          ($PXU..R$MODUL..BB) CL
DEF CL      (NAME ($PXU..R$MODUL..BB) -
             MODEL($MODEL) -
             REC (5000 5000) -
             RECSZ(140 140) -
             RECFM(FB(140)) -
             ) DATA (NAME ($PXU..R$MODUL..BB.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORDLIB
/*
/. JCLST03
* --- STEP 03 --- PTU810
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7BB,'$PXU..R$MODUL..BB',,VSAM
/* PAC7BB ON TAPE : SYS012 ASSIGNED
// DLBL PAC7ME,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTU810,SIZE=(AUTO,50K)
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | | |
|------------------------------------|------|-----|
| MANAGER'S UTILITIES | PAGE | 240 |
| GBIR: PARTITIONED DATABASE MANAGER | | 4 |
| MESN: SUB-NETWORK MERGE | | 3 |
| | | 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

GBIR: PARTITIONED DATABASE MANAGER

MESN: SUB-NETWORK MERGE

4

3

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

GBIR: PARTITIONED DATABASE MANAGER

MESN: SUB-NETWORK MERGE

4

3

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

243

4
3
5

4.3.5.3. MESN: DESCRIPTION OF STEPS

MESN: DESCRIPTION OF STEPS

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 : Physical name = \$PXU..SEQ.\$PRO\$PFI.PC

Tape is SYS015

-Extracted sub-network

PAC7BB : Physical name = \$PXU..R\$MODUL..BB

Tape is SYS011

-Error message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Transaction file:

-User input

PAC7ME : Physical name = \$XW..W\$MODUL..INPUT

.Output file:

-Merge file to be reorganized

PAC7CP : Physical name = \$PXU..SEQ.\$PRO\$PFI..NEWPC

Tape is SYS012

.Output reports:

-Merge report

SYS007

-Batch-procedure authorization option

SYS005

.Return code:

. 8 : Pas d'autorisation procédure batch

The merge result MUST BE REORGANIZED (REOR procedure) before the restoration.

MANAGER'S UTILITIES

GBIR: PARTITIONED DATABASE MANAGER

MESN: SUB-NETWORK MERGE

4

3

5

4.3.5.4. MESN: EXECUTION JCL

```

===MOD MESN
* $X JOB JNM=$PRFJ.MESN,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.MESN          ***** $PROD $PREL *****
*      LIBRARY COMPARISON UTILITY
*      SUB-NETWORK MERGE IN A BACKUP
/* *****
/* INPUT  : ONE USER AND LIBRARY LINE FOR EACH LIBRARY OF THE
/*          SUB-NETWORK TO BE MERGED (99 LINES MAXIMUM)
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI.$SLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL$) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
* $X SLI MEM=DF$PRO.$PFI.PC.$SLIT
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
R*LIB
/*
/. JCLST03
* --- STEP 03 --- PTU815
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7BB,'$PXU..REMSN.BB',,VSAM
/* PAC7BB ON TAPE : SYS011 ASSIGNED
// DLBL PAC7CP,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7CP ON TAPE : SYS012 ASSIGNED
// DLBL PAC7ME,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..PC',,VSAM
/* PAC7PC ON TAPE : SYS015 ASSIGNED
// ON $RC GT 0 GOTO $EOJ
// EXEC PTU815,SIZE=AUTO
/*
// ON $RC<16 CONTINUE
/. JCLST04
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC.$SLIT
/*
/. STEPEND
* --- STEP 05 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | |
|-------------------------|--|
| MANAGER'S UTILITIES | |
| LOAE: AE - AP RELOADING | |
| LOAE: INTRODUCTION | |

| | |
|------|-----|
| PAGE | 245 |
| | 4 |
| | 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

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
 LOAE: AE - AP RELOADING
 LOAE: EXECUTION JCL

4
 4
 4

4.4.4. LOAE: EXECUTION JCL

```

===MOD LOAE
* $X JOB JNM=$PRFJ.LOAE,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.LOAE          ***** $PROD $PREL *****
*          RELOADING OF USER PARAMETERS
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL          (NAME ($XW..W$MODUL..INPUT) -
                MODEL($MODEL$) -
                REC (1 200) -
                ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
NRREST
/*
/. JCLST03
* --- STEP 03 --- PACU80
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AP,'$PXY..$PRO.$PRO.AP',,VSAM
// DLBL PAC7CE,'$PXY..SEQ.$PRO$PRO..PE',,VSAM
// DLBL PAC7LE,'$PXY..SEQ.$PRO$PRO..AE0',,VSAM
// DLBL PAC7MC,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PACU80,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```


| | PAGE | 249 |
|---|------|-----|
| 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.

MANAGER'S UTILITIES

VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY

VINS: USER INPUT

4

5

2

4.5.2. VINS: USER INPUT

VINS: INPUT-PROCESSING-RESULTSUSER 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 prin- |
| ! ! | ! ! | ! ! | ! ting 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

VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY

VINS: DESCRIPTION OF STEPS

4

5

3

4.5.3. VINS: DESCRIPTION OF STEPS

VINS: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001DATABASE UPDATE: PACINS

.Permanent update files:

-Data file

PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

-Index file

PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN

-Journal file

PAC7AJ : Physical name = \$PXU..\$PRO.\$PFI.AJ

.Permanent input file:

-Error message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input-transaction files:

-User-Entity transactions

PAC7MV : Physical name = \$PXY..\$PRO.\$PRO.VGE

-'*' line transaction

PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output reports:

-Update report

SYS005

-Deletion-transaction list

SYS007

-Batch-procedure error report

SYS003

MANAGER'S UTILITIES

VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY

VINS: EXECUTION JCL

4

5

4

4.5.4. VINS: EXECUTION JCL

```

===SEQ FOR VISP
===MOD VINS
* $X JOB JNM=$PRFJ.VINS,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST3$PWL
/* *****
// JOB $PRFJ.VINS          ***** $PROD $PREL *****
*      UPDATING OF VISUAL ENTITIES DICTIONARY
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
// ASSGN SYS003,$LST3 3TRD PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
/*
/. JCLST03
* --- STEP 03 --- PACINS
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PX..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AJ,'$PXU..$PRO.$PFI.AJ',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM,DISP=(,DELETE,KEEP)
// DLBL PAC7MV,'$PX..$PRO.$PRO.VGE',,VSAM
// EXEC PACINS,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```

| | | |
|--|------|-----|
| MANAGER'S UTILITIES | PAGE | 253 |
| 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

4

RTLO: DESCRIPTION OF STEPS

6

2

4.6.2. RTLO: DESCRIPTION OF STEPS

RTLO: DESCRIPTION OF STEPSRETRIEVAL OF INVALID LOCKS: PTULOI

.Permanent Input files:
-Error-message file
PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Permanent Input/Output files:
-Data file
PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR
-Index file
PAC7AN : Physical name = \$PXU..\$PRO.\$PFI.AN

.Output file:
-Generated deletion transactions
PAC7MB : Physical name = \$PXU..R\$MODUL..INPUT

.Output report:
-Lists
SYS005

.Internal Sort:

MANAGER'S UTILITIES

RTLO: DELETION OF INVALID UPDATE LOCKS

4

6

RTLO: EXECUTION JCL

3

4.6.3. RTLO: EXECUTION JCL

```

===MOD RTLO
* $X JOB JNM=$PRFJ.RTLO,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.RTLO          ***** $PROD $PREL *****
*   LOCK RETRIEVAL
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE RESULT FILE      MB --- */
DEL          ($PXU..R$MODUL..MB) CL
DEF CL      (NAME ($PXU..R$MODUL..MB) -
             MODEL($MODEL$) -
             REC (8000 500) -
             RECSZ(80 80) -
             RECFM(FB(80)) -
             ) DATA (NAME ($PXU..R$MODUL..MB.D) )
/*
/. JCLST02
* --- STEP 02 --- PTULOI
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXY..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AN,'$PXU..$PRO.$PFI.AN',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$PXU..R$MODUL..MB',,VSAM
// EXEC PROC=$SORTWK CALL SORTWORK DEFINITION
// EXEC PTULOI,SIZE=(AUTO,50K)
// EXEC LISTLOG
/&
* $X EOJ

```

4.7. UXSR: PARTIAL SUB-NETWORK EXTRACTION

4.7.1. UXSR: INTRODUCTION

UXSR: INTRODUCTION

The Partial Sub-Network Extraction procedure (UXSR) creates a VisualAge Pacbase sub-network from an existing database, by:

- . Creating Libraries (MLIB equivalent)
- . Merging Libraries
- . Renaming Libraries

It is also possible to select:

- . A frozen session (nT):

This frozen session will become the current session in the new Database.

No other frozen session will be selected.

The image of this Database will be identical to the view which existed in the nT frozen session, but this time it will be in n+1 current session.

- . The current session or all sessions (current included):

Via an option, you can select all the sessions ('T' in position 67 of the * line), or only the current session (' ' in position 67 of the * line).

EXAMPLES:

- . Creation of Libraries:

```
C*CEN   AAA   (1)
C*APPCENBBB (2)
```

- (1) Creation of the CEN Library. AAA must not exist in the source Database.
- (2) Creation of the APP Library in the CEN Library. BBB must not exist in the source Database.

- . Merging of Libraries in the same Library:

```
C*CEN   CEN   (1)
C*APPCENAPP (2)
C*APPCENBQQ (2)
```


| | PAGE | 257 |
|--------------------------------------|------|-----|
| MANAGER'S UTILITIES | | 4 |
| UXSR: PARTIAL SUB-NETWORK EXTRACTION | | 7 |
| UXSR: INTRODUCTION | | 1 |

- (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

4

7

3

4.7.3. UXSR: DESCRIPTION OF STEPS

UXSR: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001FORMATTING OF THE SEQUENTIAL IMAGE: UTIXSR

.Permanent input files:

-Data file

PAC7AR : Physical name = \$PXU..\$PRO.\$PFI.AR

-Error-message file

PAC7AE : Physical name = \$PXY..\$PRO.\$PRO.AE

.Input transaction file:

-Update transactions

PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:

-Sequential image of the database

PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI.UXSR

.Output reports:

-List of user transactions

SYS007

-Resulting Database-condition

SYS007

-Batch-procedure authorization option

SYS005

MANAGER'S UTILITIES

UXSR: PARTIAL SUB-NETWORK EXTRACTION

UXSR: EXECUTION JCL

4

7

4

4.7.4. UXSR: EXECUTION JCL

```

===MOD UXSR
* $X JOB JNM=$PRFJ.UXSR,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
* $X LST CLASS=$PCL,DISP=$PDL,LST=$LST2$PWL
/* *****
// JOB $PRFJ.UXSR          ***** $PROD $PREL *****
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
// ASSGN SYS007,$LST2 2ND  PRINTER L.U.
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL       (NAME ($XW..W$MODUL..INPUT) -
              MODEL($MODEL F) -
              REC (1 200) -
              ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE RESULT FILE PC --- */
DEL          ($PXU..R$MODUL..PC) CL
DEF CL       (NAME ($PXU..R$MODUL..PC) -
              MODEL($MODEL V) -
              REC (100000 100000) -
              RECSZ(63 159) -
              ) DATA (NAME ($PXU..R$MODUL..PC.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
/*
/. JCLST03
* --- STEP 03 --- UTIXSR
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7AE,'$PXU..$PRO.$PRO.AE',,VSAM
// DLBL PAC7AR,'$PXU..$PRO.$PFI.AR',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7PC,'$PXU..R$MODUL..PC',,VSAM
/* PAC7PC ON TAPE : SYS013 ASSIGNED
// EXEC UTIXSR,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ

```

5. MIGRATIONS

| | | |
|--|------|-----|
| MIGRATIONS | PAGE | 262 |
| CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS | | 5 |
| CRYP: INTRODUCTION | | 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

CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS

5

CRYP: USER INPUT

1

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

5

CRYP: DESCRIPTION OF STEPS

1

3

5.1.3. CRYP: DESCRIPTION OF STEPS

CRYP : DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001ENCRYPTION / DECRYPTION OF PASSWORDS: PACU99

.Input files:

-User parameter backup

PAC7CE : Physical name = \$PXY..R\$MODUL..CE

-User input

PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:

-User parameter backup

PAC7EC : Physical name = \$PXY..R\$MODUL..EC

.Output report:

-Execution report

SYS005

MIGRATIONS

CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS

5

CRYP: EXECUTION JCL

1

4

5.1.4. CRYP: EXECUTION JCL

```

===MOD CRYP
* $X JOB JNM=$PRFJ.CRYP,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.CRYP          ***** $PROD $PREL *****
*   CODING-DECODING OF USER PASSWORDS
/* *****
/*   FORMAT OF TRANSACTIONS AT INPUT :
/*   .. A * USER CODE AND PASSWORD LINE
/*      (AUTHORIZATION LEVEL 4 ON PARAMETERS)
/*   .. A COMMAND LINE
/*      COL 3-8 : 'CODE '
/*           : 'DECODE'
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/* --- DELETE-DEFINE TEMPORARY FILE CE --- */
DEL          ($XW..W$MODUL..CE) CL
DEF CL      (NAME ($XW..W$MODUL..CE) -
             MODEL($MODEL) -
             REC (1000 1000) -
             RECSZ(80 80) -
             RECFM(FB(80)) -
             ) DATA (NAME ($XW..W$MODUL..CE.D) )
/* --- DELETE-DEFINE TEMPORARY FILE EC --- */
DEL          ($XW..W$MODUL..EC) CL
DEF CL      (NAME ($XW..W$MODUL..EC) -
             MODEL($MODEL) -
             REC (1000 1000) -
             RECSZ(80 80) -
             RECFM(FB(80)) -
             ) DATA (NAME ($XW..W$MODUL..EC.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
*USER      PASSWORD
*CODE
/*
/. JCLST03
* --- STEP 03 --- PACU99
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7CE,'$XW..W$MODUL..CE',,VSAM
// DLBL PAC7EC,'$XW..W$MODUL..EC',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PACU99,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
DEL          ($XW..W$MODUL..CE)      CL
DEL          ($XW..W$MODUL..EC)      CL
/*

```

MIGRATIONS

CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS

CRYP: EXECUTION JCL

5

1

4

// EXEC LISTLOG

/&

* \$X EOJ

| | PAGE | 267 |
|---|------|-----|
| MIGRATIONS | | |
| 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

MIGRATIONS

LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE

5

LVBL: DESCRIPTION OF STEPS

2

2

5.2.2. LVBL: DESCRIPTION OF STEPS

LVBL: DESCRIPTION OF STEPSREPLACEMENT OF LOW-VALUES WITH BLANKS: PTULVB

.Input file:

-Database backup

PAC7MC : Physical name = \$PXU..SEQ.\$PRO\$PFI.PC
Tape is SYS011

.Output file:

-Database backup

PAC7PC : Physical name = \$PXU..SEQ.\$PRO\$PFI.NEWPC
Tape is SYS012

MIGRATIONS

LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE
 LVBL: EXECUTION JCL

5
 2
 3

5.2.3. LVBL: EXECUTION JCL

```

===MOD LVBL
* $X JOB JNM=$PRFJ.LVBL,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.LVBL          ***** $PROD $PREL *****
* REPLACING LOW-VALUES BY SPACES IN A PC FILE
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..SSLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=DF$PRO.$PFI.PC..SSLIT
/*
/. JCLST02
* --- STEP 02 --- PTULVB
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MC,'$PXU..SEQ.$PRO$PFI..PC',,VSAM
// DLBL PAC7PC,'$PXU..SEQ.$PRO$PFI..NEWPC',,VSAM
/* PAC7PC ON TAPE : SYS012 ASSIGNED
// EXEC PTULVB,SIZE=AUTO
/. STEPEND
* --- STEP 03 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
* $X SLI MEM=RG$PRO.$PFI.PC..SSLIT
/*
// EXEC LISTLOG
/&
* $X EOJ

```

| | PAGE | 270 |
|--|------|-----|
| 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

SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION

5

3

SMTD: DESCRIPTION OF STEPS

2

5.3.2. SMTD: DESCRIPTION OF STEPS

SMTD: DESCRIPTION OF STEPSTD BACKUP: PTASVD

.Permanent input file:

-Table-description file

PAC7TD : Physical name = \$TXU..\$TRO.00TD

.Output file:

-Table-description backup for migration

PAC7TC : Physical name = \$TXU..R\$MODUL..TC

Tape is SYS012

MIGRATIONS

SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION
 SMTD: EXECUTION JCL

5
 3
 3

5.3.3. SMTD: EXECUTION JCL

```

===SEQ FOR TAB
===MOD SMTD
* $X JOB JNM=$PRFJ.SMTD,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.SMTD          ***** $PROD $PREL *****
*   BACKUP OF TD FILE FOR MIGRATION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE RESULT FILE          TC --- */
DEL              ($PXU..R$MODUL..TC) CL
DEF CL           (NAME ($PXU..R$MODUL..TC) -
                  MODEL($MODEL) -
                  REC (1000 500) -
                  RECSZ(80 80) -
                  RECFM(FB(80)) -
                  ) DATA (NAME ($PXU..R$MODUL..TC.D) )
/*
/. JCLST02
* --- STEP 02 --- PTASVD
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7TC,'$TXU..R$MODUL..TC',,VSAM
/* PAC7TC ON TAPE : SYS012 ASSIGNED
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// EXEC PTASVD,SIZE=AUTO
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```


| | | |
|---|------|-----|
| MIGRATIONS | PAGE | 273 |
| 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

RMTD: RESTORATION OF TABLE DESCRIPTIONS

5

RMTD: DESCRIPTION OF STEPS

4

2

5.4.2. RMTD: DESCRIPTION OF STEPS

RMTD: DESCRIPTION OF STEPSTD FILE RESTORATION: PTARSD

.Input backup file:

-Table-Description sequential file

PAC7TC : Physical name = \$TXU..R\$MODUL..TC

Tape is SYS012

.Output file:

-Table-Description file

PAC7TD : Physical name = \$TXU..\$TRO.00TD

MIGRATIONS

RMTD: RESTORATION OF TABLE DESCRIPTIONS

5

RMTD: EXECUTION JCL

4

3

5.4.3. RMTD: EXECUTION JCL

```
===SEQ FOR TAB
===MOD RMTD
* $X JOB JNM=$PRFJ.RMTD,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.RMTD          ***** $PROD $PREL *****
*   RESTORE TD FILE AFTER MIGRATION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- PTARSD
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7TC,'$TXU..RSMTD.TC',,VSAM
/* PAC7TC ON TAPE : SYS012 ASSIGNED
// DLBL PAC7TD,'$TXU..$TRO.00TD',,VSAM
// EXEC PTARSD,SIZE=AUTO
// EXEC LISTLOG
/&
* $X EOJ
===SEQ
```

| | | |
|------------------------------------|------|-----|
| MIGRATIONS | PAGE | 276 |
| RPTD: TABLE DESCRIPTIONS RETRIEVAL | | 5 |
| RPTD: INTRODUCTION | | 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

5

RPTD: USER INPUT

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

5

RPTD: DESCRIPTION OF STEPS

5

3

5.5.3. RPTD: DESCRIPTION OF STEPS

RPTD : DESCRIPTION DES ETAPES2.0 RETRIEVAL OF TD FILE: PTAR20

.Input files:

-Table-descriptions backup

PAC7TC : Physical name = \$TXU..\$OLDTDSAV

-User parameter-line

PAC7MB : Physical name = \$XW..W\$MODUL..INPUT

.Output file:

-2.0 backup of table-descriptions

PAC7TR : Physical name = \$TXU..\$TDSAV

.Output report:

-Retrieval report

SYS005

MIGRATIONS

RPTD: TABLE DESCRIPTIONS RETRIEVAL

5

RPTD: EXECUTION JCL

5

4

5.5.4. RPTD: EXECUTION JCL

```

===SEQ FOR TAB
===MOD RPTD
* $X JOB JNM=$PRFJ.RPTD,CLASS=$PCJ,DISP=$PDJ,USER='$PUJ'$PWJ
* $X LST CLASS=$PCL,DISP=$PDL$PWL
/* *****
// JOB $PRFJ.RPTD          ***** $PROD $PREL *****
*          RETRIEVAL OF TABLE DESCRIPTION FOR MIGRATION
/* *****
// OPTION $OPT
* $X SLI MEM=DBJB$PRO.$PFI..$SLIT
/* GOTO JCLSTXX
/. JCLST01
* --- STEP 01 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
/* --- DELETE-DEFINE 'INPUT' TEMPORARY FILE --- */
DEL          ($XW..W$MODUL..INPUT) CL
DEF CL      (NAME ($XW..W$MODUL..INPUT) -
             MODEL($MODEL) -
             REC (1 200) -
             ) DATA (NAME ($XW..W$MODUL..INPUT.D) )
/*
/. JCLST02
* --- STEP 02 --- PTU001
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// EXEC PTU001,SIZE=AUTO
AA
/*
/. JCLST03
* --- STEP 03 --- PTAR20
// DLBL IJSYSUC,'$VCAP',,VSAM
// DLBL PAC7MB,'$XW..W$MODUL..INPUT',,VSAM
// DLBL PAC7TD,'$TXU..$OLDTDSAV',,VSAM
// DLBL PAC7TR,'$TXU..TDSAV',,VSAM
// EXEC PTAR20,SIZE=AUTO
/. STEPEND
* --- STEP 04 --- IDCAMS
// DLBL IJSYSUC,'$VCAP',,VSAM
// EXEC IDCAMS,SIZE=AUTO
DEL          ($XW..W$MODUL..INPUT) CL
/*
// EXEC LISTLOG
/&
* $X EOJ
===SEQ

```