

VisualAge Pacbase



# The Administrator's Procedures IMS/VS

*Version 3.0*





VisualAge Pacbase



# The Administrator's Procedures IMS/VS

*Version 3.0*

#### Note

Before using this document, read the general information under “Notices” on page vii.

According to your licence 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.ibm.com/software/ad/vapacbase/productinfo.htm>

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

#### First Edition (March 2003)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.0

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at: <http://www.ibm.com/software/ad/vapacbase/support.htm> or to the following postal address:

IBM Paris Laboratory  
1, place Jean-Baptiste Clément  
93881 Noisy-le-Grand, 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 1983,2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Notices</b> . . . . .	<b>vii</b>	Sub-Network Backup . . . . .	49
<b>Trademarks</b> . . . . .	<b>ix</b>	SASN - Introduction . . . . .	49
<b>Chapter 1. Overview</b> . . . . .	<b>1</b>	SASN - User Input . . . . .	49
Presentation of the Manual . . . . .	1	Partial Sub-Network Extraction . . . . .	50
Presentation of the Procedures . . . . .	1	UXSR - Introduction . . . . .	50
User Identification . . . . .	2	UXSR - User Input . . . . .	51
Access Authorization . . . . .	3	PACS - Description of Steps . . . . .	52
Abnormal Endings . . . . .	3	PACS - Execution JCL . . . . .	54
<b>Chapter 2. Administration Database Management</b> . . . . .	<b>5</b>	UPDT - Freeze . . . . .	58
ARAD - Archival . . . . .	5	UPDT - Introduction . . . . .	58
ARAD - Introduction . . . . .	5	UPDT - Input . . . . .	59
ARAD - Input / Processing / Results . . . . .	5	UPDT - Description of Steps . . . . .	59
ARAD - Description of Steps . . . . .	8	UPDT - Execution JCL . . . . .	61
ARAD - Execution JCL . . . . .	10	SASY - Database System Backup Complement . . . . .	64
PACG - Backup . . . . .	12	SASY - Introduction . . . . .	64
PACG - Introduction . . . . .	12	SASY - Description of Steps . . . . .	65
PACG - Input / Processing / Results . . . . .	13	SASY - Execution JCL . . . . .	65
PACG - Description of Steps . . . . .	14	REST - Database Restoration . . . . .	67
PACG - Execution JCL . . . . .	15	REST - Introduction . . . . .	67
ROAD - Reorganization . . . . .	17	REST - Input / Processing / Results . . . . .	68
ROAD - Introduction . . . . .	17	REST - Description of Steps . . . . .	71
ROAD - Input / Processing / Results . . . . .	18	REST - Execution JCL . . . . .	74
ROAD - Description of Steps . . . . .	20	RESY - Database System Restoration Complement . . . . .	80
ROAD - Execution JCL . . . . .	25	RESY - Introduction . . . . .	80
RSAD - Restoration . . . . .	31	RESY - Input / Processing / Results . . . . .	81
RSAD - Introduction . . . . .	31	RESY - Description of Steps . . . . .	83
RSAD - Input / Processing / Results . . . . .	32	RESY - Execution JCL . . . . .	86
RSAD - Description of Steps . . . . .	33	ARCH - Journal Archival . . . . .	91
RSAD - Execution JCL . . . . .	37	ARCH - Introduction . . . . .	91
<b>Chapter 3. Database Management Utilities</b> <b>43</b>	<b>43</b>	ARCH - Input / Processing / Results . . . . .	91
Backup Procedures . . . . .	43	ARCH - Description of Steps . . . . .	94
Introduction . . . . .	43	ARCH - Execution JCL . . . . .	96
PACS - Input Common to Managers . . . . .	43	REOR - Reorganization . . . . .	99
Database Management . . . . .	44	REOR - Introduction . . . . .	99
MLIB - Introduction . . . . .	44	REOR - Input / Processing / Results . . . . .	100
MLIB - Input / Processing / Results . . . . .	44	REOR - Description of Steps . . . . .	103
Database Backup . . . . .	47	REOR - Execution JCL . . . . .	108
SAVE - Introduction . . . . .	47	<b>Chapter 4. Manager's Utilities</b> . . . . .	<b>117</b>
SAVE - Input / Processing / Results . . . . .	48	PACX - Extractions . . . . .	117
		PACX - Introduction . . . . .	117
		PACX - Input Common to Extractors . . . . .	117
		Extraction of Archived Transactions . . . . .	119
		EXPJ - Introduction . . . . .	119

EXPJ - User Input . . . . .	119	Analysis . . . . .	171
Extraction of Libraries . . . . .	121	PQCA - Introduction . . . . .	171
EXLI - Introduction . . . . .	121	PQCA - User Input . . . . .	172
EXLI - User Input . . . . .	121	PQCA - Description of Steps . . . . .	172
Extraction for Purge . . . . .	122	PQCA - Execution JCL . . . . .	174
EXPU - Introduction . . . . .	122	Extraction of Quality Rules . . . . .	177
EXPU - User Input . . . . .	123	PQCE - Introduction . . . . .	177
Standardization Utility . . . . .	125	PQCE - Input / Processing / Results	178
RMEN - Introduction . . . . .	125	PQCE - Description of Steps . . . . .	179
RMEN - User Input . . . . .	126	PQCE - Execution JCL . . . . .	182
RMEN - Recommendations and Restrictions . . . . .	130	<b>Chapter 6. Versioning Facilities . . . . .</b>	<b>187</b>
Sub-Network and Entities Comparison	133	Standard Bridge (PCM) . . . . .	187
CPSN - Introduction . . . . .	133	Introduction . . . . .	187
CPSN - User Input . . . . .	134	Database Automatic Freeze . . . . .	187
PACX - Description of Steps . . . . .	134	HIPM - Introduction . . . . .	187
PACX - Execution JCL . . . . .	136	HIPM - Input / Processing / Results	188
Session Management . . . . .	140	HIPM - Description of Steps . . . . .	189
Introduction . . . . .	140	HIPM - Execution JCL . . . . .	190
ESES - Session Numbers Extraction . . . . .	140	Generation Simulation . . . . .	192
ESES - Introduction . . . . .	140	SIPM - Introduction . . . . .	192
ESES - User Input . . . . .	141	SIPM - Input / Processing / Results	192
ESES - Description of Steps . . . . .	141	SIPM - Description of Steps . . . . .	194
ESES - Execution JCL . . . . .	142	SIPM - Execution JCL . . . . .	195
CSES - Compression of Session Numbers	143	Extraction of the Development Database	
CSES - Introduction . . . . .	143	Data . . . . .	197
CSES - User Input . . . . .	144	EXPM - Introduction . . . . .	197
CSES - Description of Steps . . . . .	144	EXPM - Input / Processing / Results	198
CSES - Execution JCL . . . . .	145	EXPM - Description of Steps . . . . .	199
Storage Optimization of Multi-Volumes . . . . .	148	EXPM - Execution JCL . . . . .	201
STOP - Introduction . . . . .	148	Comparison with Extracted Files . . . . .	203
STOP - User Input . . . . .	148	CPPM - Introduction . . . . .	203
STOP - Description of Steps . . . . .	149	CPPM - Input / Processing / Results	204
STOP - Execution JCL . . . . .	150	CPPM - User File . . . . .	204
Database statistics . . . . .	152	CPPM - Description of Steps . . . . .	206
STAT - Introduction . . . . .	152	CPPM - Execution JCL . . . . .	208
STAT - User input . . . . .	154	Integrity Control of Events/Elements . . . . .	211
STAT - Description of steps . . . . .	154	CHPM - Introduction . . . . .	211
STAT - Execution JCL . . . . .	155	CHPM - Input / Processing / Results	211
<b>Chapter 5. Analysis of Activity and Quality Control . . . . .</b>	<b>157</b>	CHPM - Description of Steps . . . . .	212
Analysis of Activity . . . . .	157	CHPM - Execution JCL . . . . .	212
ACTI - Introduction . . . . .	157	Update . . . . .	215
ACTI - Query Language . . . . .	158	UPPM - Introduction . . . . .	215
ACTI - User Input . . . . .	167	UPPM - Input / Processing / Results	215
ACTI - Description of Steps . . . . .	167	UPPM - Description of Steps . . . . .	215
ACTI - Execution JCL . . . . .	168	UPPM - Execution JCL . . . . .	216
Pacbench Quality Control . . . . .	171	Transactions Archiving . . . . .	217
Introduction . . . . .	171	ARPM - Introduction . . . . .	217
		ARPM - Input / Processing / Results	218
		ARPM - Description of Steps . . . . .	218

ARPM - Execution JCL . . . . .	220	Creation of the Transfer File . . . . .	251
Pac/Transfer . . . . .	224	TRPF - Introduction . . . . .	251
Introduction . . . . .	224	TRPF - User Input . . . . .	251
Processes Chronology . . . . .	225	TRPF - Description of Steps . . . . .	252
Update of Transfer Parameters . . . . .	226	TRPF - Execution JCL . . . . .	254
TRUP - Introduction . . . . .	226	Preparing DSMS Environment . . . . .	258
TRUP - User Input . . . . .	228	TRDU - Introduction . . . . .	258
TRUP - Description of Steps . . . . .	232	TRDU - User Input . . . . .	260
TRUP - Execution JCL . . . . .	234	TRDU - Description of Steps . . . . .	260
Print of Transfer Parameters . . . . .	238	TRDU - Execution JCL . . . . .	264
TRED - Introduction . . . . .	238	DSMS Update Before Database Update	270
TRED - User Input . . . . .	238	TRRP - Generation of Transfer	
TRED - Description of Steps . . . . .	239	Transactions . . . . .	271
TRED - Execution JCL . . . . .	240	TRRP - Introduction . . . . .	271
Compression of Archived Journal . . . . .	244	TRRP - User Input . . . . .	273
TRJC - Introduction . . . . .	244	TRRP - Description of Steps . . . . .	274
TRJC - User Input . . . . .	245	TRRP - Execution JCL . . . . .	276
TRJC - Description of Steps . . . . .	245	Update of the Development Database . . . . .	280
TRJC - Execution JCL . . . . .	247	Reinitialization of DSMS Environment	280





---

## 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504-1785, U.S.A.

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 the information which has been exchanged, should contact IBM Paris Laboratory, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex. 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.



---

# Chapter 1. Overview

---

## Presentation of the Manual

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

These procedures are related mainly to the following operations fields:

- Management of the administration Database,
- Administration of the development Databases,
- Manager's utilities,
- Analysis of activity and quality control,
- Management of versions.

---

## Presentation of the Procedures

Batch processes are grouped into procedures. The objective of the following chapters is to present each of the procedures that are likely to be used, and to specify their execution conditions.

The following elements are included for each procedure:

- a general introduction including:
  - the Execution Conditions,
  - operations to be performed in case of Abnormal Executions.
- the description of the User Input, Processes and Results obtained, possibly including use recommendations.
- the Description of Steps.

To use a procedure on a given Database, the user must have the corresponding authorization.

Each user has:

- a general level of authorizations to the batch procedures,
- a specific authorization level per Database

User authorizations are defined in the Administration Database.

## User Identification

Batch procedures which access the Databases 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.

Some information entered on this line 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.

Position	Length	Value	Meaning
2	1	*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code
22	4	nnnn	Session number
26	1	T	Test session
		H	Frozen session
27	1		With the UPDT procedure in case of multiple deletion:
		N	Print all transactions, including generated transactions (default option)
		O	Print transactions entered by the user and erroneous generated transactions
		E	Print erroneous transactions only
			The 2 following fields must be valued for all extraction procedures generating update transactions which will modify a Library/session under DSMS control (you can also value them on the UPDT '*' line,
40	3		Product code (3 character-code),
43	6		Change number (6 character-code, non-significant zeros must be entered),
			These two codes will appear in the Journal after the execution of UPDT
49	1		Transfer of Entity Lock:

Position	Length	Value	Meaning
		blank	Replacement of the user code which locks the entity with the '*' line
		1	New entities created from the extracted entities are not locked after the execution of UPDT
		2	The user code which locks the entity is kept
50	1		Transfer of the password on the extraction procedures, on the '*' line of output transactions
		blank	The password is not transferred in the output file,
		1	The password is transferred, (Note : for EXTR, the '*' line is transferred in the output file only if you have entered a 'C' in Column 1)

---

## Access Authorization

An '\*' line with a user code and password is required by all procedures.

The Administrator manages the user access authorizations on batch procedures via the Administrator workbench.

---

## Abnormal Endings

Input-output errors on the system files or on the database provoke an abnormal end with an abort (code 12 in the concerned program, switch 20 in the procedure) accompanied by a message on the SYSOUT file.

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

```
VisualAge Pacbase Debug information
  Error on file:FF Oper:-- Key:xxxxx Cause: nn
  Program:BVPppppp Date:dd/dd/dddd 11:42:32 Lib:xxx Sess: xx
-----
```

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

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



---

## Chapter 2. Administration Database Management

---

### ARAD - Archival

#### ARAD - Introduction

This procedure saves the Journal file as a sequential file, and re-initializes it both logically and physically.

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

The archived transactions 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 CONDITIONS

On-line access must be closed.

#### ABNORMAL EXECUTIONS

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

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

#### NOTE

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

#### ARAD - Input / Processing / Results

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

This procedure includes specific optional input to:

- Deactivate previously archived transactions that are considered obsolete.

- Signal the absence of previously archived transactions in input.
- Signal the unavailability of the Data file (AR) in input.
- Request only the re-initialization of the transaction file.

The structure of this input is as follows:

Position	Length	Value	Meaning
2	1	'S'	Line code
3	4	nnnn	Session number
7	8	ccyymmdd	OR date up to which the user requests deactivation
15	1	'I'	Absence of previously archived transactions
16	1	'D'	Data file unavailable
17	1	'J'	Re-initialization without backup, the transactions already archived are NOT retrieved in output.

The session number and the date are exclusive. They are ignored if the absence of input transactions is signalled (refer to Section Recommendations).

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

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

### WARNING

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

If an error occurs on one of the options, a 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, some information in the Specifications Dictionary is sometimes lost, thus preventing the execution of the archiving and restoration 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 backup procedure will not take the Data file (AR) into account. However, the restoration procedure must be executed afterward, since under these conditions, the backup 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 backup procedure formats an empty Journal file. The restoration procedure may then be executed (not compulsory). In this case, the content of the journal file (AJ) is lost.
- If the transactions Back-up file is lost or destroyed, an 'T' must be entered in column 15. As a result, the archiving procedure reformats a new sequential backup file of (archived) transactions.

If one of these columns is accidentally set, and if the archiving procedure is executed while 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 (-1) and (0) to obtain (+1).
- 'D' in col. 16: The archiving 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, (+1 version 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 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 session number.

### **ARAD - Description of Steps**

INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

ARCHIVAL OF JOURNAL FILE: PTU300

This step:

- writes obsolete transactions to be purged on 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.

<b>Code</b>	<b>Physical name</b>	<b>Type</b>	<b>Label</b>
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
PAC7JP	&INDSN..BVPPK(0)	Input	Previously archived transactions
BVP8GJ	&INDSV..BVP GJ	Input	Journal to reinitialize from administration database
PAC7MB	&&ARADMB	Input	User transaction
PAC7BM	&&ARADB M	Output	User transaction
BVP8GR	&INDSV..BVP GR	Input/Output	Development Database data
PAC7PJ	&INDSN..BVPPK(+1)	Output	Archived update transactions
PAC7PQ	DUMMY	Output	Deactivated transactions (length=170) : modify file name to save deactivated transactions
PAC7EU		Report	Output report
PAC7DD		Report	Batch procedures authorization option

Return code:

- 0: No error detected on the files.
- 4: Error in journal record (Date or session number not numeric).

- 8: No access authorization for batch procedure OR invalid database (in this case, restart the procedure with 'D' in column 16 of the user input ).
- 12: Input-output error on a file.

#### CREATION OF THE JOURNAL FILE: IDCAMS - IEBFR14

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

Code	Physical name	Type	Label
BVP8GJ	&INDSV..BVPGJ	Output	Administration database Journal

#### RE-INITIALIZATION OF THE JOURNAL FILE: PTU320

This step executes two types of operations:

- Creates the first record in the Journal file,
- Re-initializes the data file flag with the Journal file's address.

Code	Physical name	Type	Label
PAC7BM	&&ARADBM	Input	User transaction
BVP8GR	&INDSV..BVPGR	Input/Output	Administration Database data
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP8GJ	&INDSV..BVPGJ	Output	Journal file to re-initialize
PAC7EU		Report	Review of re-initialization

Return codes:

- 0: No error detected
- 8: the Database is invalid

If the archival and backup procedures are grouped into one job, the PTU320 return code can be tested in order to condition the execution of the backup procedure.

#### CANCELLATION OF BACKUP IN CASE OF INCONSISTENCY: IEFBR14

This step is executed if the return code of the archiving program (PTU300) is other than 0.

Code	Physical name	Type	Label
DDPK	&INDSN..BVPPK(+1)	Input	Updated archived transactions

## ARAD - Execution JCL

```
//* -----
//*      VISUALAGE PACBASE
//*
//* -----
//*      - ARCHIVAL OF THE JOURNAL -
//*      DATABASE ADMINISTRATOR
//* -----
//*
//* INPUT      : COMMAND FOR DEACTIVATION OF ARCHIVED
//*            TRANSACTION
//* COL 2      : "S"
//* COL 3 TO 6 : SESSION NUMBER
//* COL 7 TO 14 : DATE (CCYYMMDD)
//* COL 15     : " " PRESENCE OF ARCHIVED TRANSACTION FILE
//*           : "I" ABSENCE OF ARCHIVED TRANSACTION FILE
//* COL 16     : " " PRESENCE OF DATA FILE (AR)
//*           : "D" ABSENCE OF DATA FILE (AR)
//* COL 17     : " " ARCHIVAL AND REINITIALIZATION
//*           : "J" REINITIALIZATION WITHOUT ARCHIVAL
//*
//* IN THE ABSENCE OF INPUT (OR ERROR ON A COMMAND PARAM.)
//* NO DEACTIVATION WILL TAKE PLACE, HOWEVER ARCHIVAL AND
//* REINITIALIZATION WILL BE EXECUTED NORMALLY.
//*
//* TRANSACTIONS WHOSE SESSION (DATE) IS PRIOR OR EQUAL TO
//* THE SESSION (DATE) INDICATED ARE NOT KEPT. THEY ARE
//* RECOVERED IN THE FILE OF DEACTIVATED TRANSACTION.
//*
//* -----
//BVPARAD  PROC INDSV='$INDSV',  INDEX OF SYSTEM VSAM FILES
//*:      SYSCAT='$VCAT',      VSAM SYSTEM CATALOG
//*:      VSAMCAT='$SCAT',     VSAM USER CATALOG
//      OUT='$OUT',           OUTPUT CLASS
//      VOLS='$VOLUN',        VOLUME OF ARCHIVED TRANSACTIONS
//      INDSN='$INDSN',       INDEX OF SYSTEM NO VSAM FILES
//      UNITS='$UNITUN',      TRANSACTION FILE UNIT
//      STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//      DSCB='$DSCB',         DSCB MODEL FILE
//      SPAPK='(TRK,(020,10),RLSE)', SPACE PK
//      SPAGJ='(TRK,(020,10),RLSE)', SPACE GJ
//      PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//      DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//      RESLIB='$RESLIB',     IMS RESLIB
//      PROCLIB='$PRCLIB',    IMS PROCLIB
//      UWK=$UWK,             WORK UNIT
//      BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//      CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&ARADMB,DISP=(,PASS),
//         UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
```

```

//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGJ DD DSN=&INDSV..BVPGJ,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFGJ),DISP=SHR
//      DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//*-----
//PTU300 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU300,GTU300$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//      &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GJ DD DSN=&INDSV..BVP8GJ,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7JP DD DSN=&INDSN..BVPPK(0),DISP=OLD
//PAC7PJ DD DSN=&INDSN..BVPPK(+1),DISP=(,CATLG,DELETE),
//      UNIT=&UNITS,VOL=SER=&VOLS,
//      SPACE=&SPAPK,
//      DCB=(&DSCB,RECFM=FB,LRECL=170,BLKSIZE=27880)
//PAC7MB DD DSN=&&ARADMB,DISP=(OLD,PASS)
//PAC7BM DD DSN=&&ARADBM,DISP=(,PASS),
//      UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7PQ DD DUMMY,DCB=BLKSIZE=170
//PAC7DD DD SYSOUT=&OUT
//PAC7EU DD SYSOUT=&OUT
//*-----

```

```

//OSAM1 EXEC PGM=IDCAMS,COND=(04,LT,PTU300)
//SYSIN DD DSN=&INDSN..BVPSY(DLBVPGJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAM2 EXEC PGM=IEFBR14,COND=(4,LT,PTU300)
//BVP8GJ DD DSN=&INDSV..BVPGJ,UNIT=&UNITS,
// DISP=(,CATLG,DELETE),VOL=SER=&VOLS,
// DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
// SPACE=&SPAGJ
//*-----
//PTU320 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU320,GTU320$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=(04,LT,PTU300)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GJ DD DSN=&INDSV..BVP8GJ,DISP=SHR
//PAC7BM DD DSN=&&ARADBM,DISP=(OLD,DELETE)
//PAC7EU DD SYSOUT=&OUT
//*-----
//DELPK EXEC PGM=IEFBR14,COND=(8,GT,PTU300)
//DDPK DD DSN=&INDSN..BVPPK(+1),DISP=(OLD,DELETE)
//

```

---

## PACG - Backup

### PACG - Introduction

The purpose of this procedure is to save the main files of the administration Database as a 'PE' sequential file.

The following files are saved:



- the data file (GR),
- the index file (GN).
- the extension data file (GY).

### EXECUTION CONDITIONS

On-line access must be prohibited.

### ABNORMAL EXECUTION

Refer to subchapter 'Abnormal Endings', chapter 'Overview'.

The main reason for an abend is that on-line environment is still open to transactions.

The procedure can therefore be restarted once the on-line environment is closed.

### ARCHIVAL AND BACKUP LINKING

If the backup procedure is preceded by a Journal archival procedure (ARAD), its execution may be conditioned by the return code of the PTU320 program of the ARAD procedure:

- 0: No error detected
- 8: Database not available

### PRINTED REPORT

Once the 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 the number of records per library and per line-type,
  - a report listing database limits reached.

## **PACG - Input / Processing / Results**

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
29	4	'SAVE'	Function code

## PACG - Description of Steps

INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

FORMATTING OF THE SEQUENTIAL IMAGE: PTU520

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
BVP8GY	&INDSV..BVPGY	Input	Administration Database extension data
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
PAC7MB	&&PACGMB	Input	Update transactions
PAC7PC	&INDSN..BVPPE(+1)	Output	Sequential image of the administration Database
PAC7RP	&&PACSRP	Output	Sequential image of data (length=153) (should be able to contain all the data)
PAC7NA	&&PACSNA	Output	Sequential image of indexes (length=59) (should be able to contain all the indexes)
PAC7NB	&&PACSNB	Output	Image of non-sorted indexes (length=59)
PAC7RY	&&PACSRY	Output	Sequential image of long data (length=1,019)
PAC7RQ	&&PACSRQ	Output	Intermediate storage (1 record, length=153)
PAC7EV		Report	User transactions list
PAC7EU		Report	Status of network before and after
PAC7EW		Report	Backup report
PAC7DD		Report	Abend output report

Return code:

- 8 : Inconsistency of the database or no authorization for batch procedure

Response to return code:

If the return code is greater than 2, the resulting backup is deleted by the next step in the procedure and a restoration must be performed using the last valid backup.

If there is no other backup to restore the database, the user should first examine the problem with the support team of the product, then the inconsistent database should be saved by the same procedure with the backup deletion step inactive. The resulting backup contains only data and can only be used after running the reorganization procedure.

#### BACKUP DELETION IN CASE OF DATABASE INCONSISTENCY: IEFBR14

This step is executed if the return code of one preceding program is greater than 2.

Code	Physical name	Type	Label
DDPE	&INDSN..BVPPE(+1)	Input	Sequential image of the administration Database

### PACG - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - BACKUP OF THE ADMINISTRATION DATABASE -
/**
/** -----
/**
//BVPPACG  PROC INDSN='$INDSN',  INDEX OF SYSTEM NO VSAM FILES
//          INDSV='$INDSV',      INDEX OF SYSTEM VSAM FILES
/**:      SYSTCAT='$VCAT',        VSAM SYSTEM CATALOG
/**:      VSAMCAT='$SCAT',        VSAM USER CATALOG
//          OUT='$OUT',           OUTPUT CLASS
//          VOLS='$SER=$VOLUN',    BACKUP VOLUME
//          UNITS='$UNITUN',      BACKUP UNIT
//          SPAPE=(TRK,(100,10),RLSE)',  SPACE BASE VA PACBASE
//          SPAMB=(TRK,(1,1),RLSE)',    SPACE OF INPUT CARD
//          SPAAN=(TRK,(700,20),RLSE)',  SPACE TEMPORARY INDEX
//          SPAAR=(TRK,(700,20),RLSE)',  SPACE TEMPORARY DATAX
//          UWK='$UWK',           WORK UNIT
//          DSCB='$DSCB',         DSCB MODEL FILE
//          STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULE
//          PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//          RESLIB='$RESLIB',       IMS RESLIB

```

```

//          PROCLIB='$PRCLIB', IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&PACGMB,DISP=(,PASS),UNIT=&UWK,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//CARTE DD DDNAME=SYSIN
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGY DD DSN=&INDSV..BVPGY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//*-----
//PTU520 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU520,GTU520$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GY DD DSN=&INDSV..BVP8GY,DISP=SHR
//PAC7MB DD DSN=&&PACGMB,DISP=(OLD,PASS)
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR

```

```

//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7PC DD DSN=&INDSN..BVPPE(+1),DISP=(,CATLG,DELETE),
//
// UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPE,
//
// DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//PAC7PD DD DUMMY
//PAC7PY DD DUMMY
//PAC7NA DD DSN=&&PACSNA,DISP=(,PASS,DELETE),
//
// UNIT=&UWK,SPACE=&SPAAN,
//
// DCB=(RECFM=FB,LRECL=59,BLKSIZE=27966)
//PAC7NB DD DSN=&&PACSNB,DISP=(,PASS,DELETE),
//
// UNIT=&UWK,SPACE=&SPAAN,
//
// DCB=(RECFM=FB,LRECL=59,BLKSIZE=27966)
//PAC7RP DD DSN=&&PACSRP,DISP=(,PASS,DELETE),
//
// UNIT=&UWK,SPACE=&SPAAR,
//
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7RQ DD DSN=&&PACSRQ,DISP=(,PASS,DELETE),
//
// UNIT=&UWK,SPACE=(TRK,1,RLSE),
//
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=153)
//PAC7RY DD DSN=&&PACSRY,DISP=(,PASS,DELETE),
//
// UNIT=&UWK,SPACE=&SPAPE,
//
// DCB=(RECFM=FB,LRECL=1019,BLKSIZE=27513)
//PAC7EU DD SYSOUT=&OUT
//PAC7EW DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7EV DD SYSOUT=&OUT
//*-----
//DELPE EXEC PGM=IEFBR14,COND=(2,GT,PTU520)
//DDPE DD DSN=&INDSN..BVPPE(+1),DISP=(OLD,DELETE)
//

```

---

## ROAD - Reorganization

### ROAD - Introduction

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

It uses the administration Database backup file and rebuilt a sequential image. This resulting image file must then be restored via the RSAD procedure.

To keep users and profiles updated since the last backup, the Administration database backup procedure must be first executed.

The operating principle 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.

This 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 Index file).
- When the administrator wants to purge the entities not used in the database.

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

### EXECUTION CONDITIONS

The administration Database may remain open during the 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.

### ABNORMAL EXECUTION

Refer to subchapter 'Abnormal Endings', chapter 'Overview'.

As specified in the recommendations below, it is 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.

## **ROAD - Input / Processing / Results**

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

Specific user input for the procedure (optional), which enables to specify:

- the 'unformatted'-type data to be purged,
- a printout of the duplicate indexes of the reorganization procedure.

Pos.	Len.	Value	Meaning
1	1	'P'	Physical purge of 'unformatted' data
3			Entity (required)
	1		Type
	2		UE call code (if '\$' type)
6	30		Code of the entity to be purged (required, full identifier)

Pos.	Len.	Value	Meaning
36	2		Type of the entity description (required, value '00' not authorized)
		'nn'	Purge of the 'nn' descriptions of the entity
		'**'	Purge of all the descriptions of the entity
38	3	'***'	Library code (required)
41	1	'N'	Flag for the purge of lower-level Libraries
42	4	'****'	Session lower bound
46	4	'****'	Session upper bound

One line per 'unformatted'-type entity

The execution of the ROAD procedure processes a maximum of 2500 instances of entities other than User Entity, and a maximum of 1000 User Entity instances.

Position	Length	Value	Meaning
2	1	'D'	Report of the list of the duplicate indexes of the REOR procedure
3	1	' '	No report of duplicate indexes
		'1'	Report of duplicate indexes
		'2'	Report of PMS calls which may be incoherent
		'3'	Reports of values '1' and '2'

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

#### ESTIMATED 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 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 'Internal report' reserving their use to the product support team in case of problem.

### 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 will be added by the database restoration procedure.

### IMPORTANT RECOMMENDATIONS

The Reorganization procedure presents a number of characteristics which the user should be aware of:

- The step that rebuilds the Index file uses a large amount of CPU time.
- 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.

## **ROAD - Description of Steps**

### INPUT RECOGNITION: PTU001

### INPUT CHECK: PTU2CL

This step checks user input and sets a return code if errors have been detected.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error labels
BVP7GN	&INDSV..BVPGN	Input	Administration database index
BVP7GR	&INDSV..BVPGR	Input	Administration database data
BVP7GU	&INDSV..BVPGU	Input	Administration database users



Code	Physical name	Type	Label
PAC7PC	&INDSN..BVPPE(0)	Input	Sequential image of the administration Database
PAC7MB	&&ROADMB	Input	Input work file
PAC7BM	&&ROADBM	Output	Formatted records
PAC7PU	&&PU	Output	Entity purge transactions (length=44)
PAC7EE		Report	Validation report
PAC7DD		Report	Batch procedure authorization option
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0: OK
- 4: Error on user inputs
- 8: No authorization for batch procedure

#### CREATION OF EXTENSION DATA FILE: IDCAMS

This step executes a DELETE/DEFINE of the work file which contains the WKROAD extension data.

Code	Physical name	Type	Label
	&INDSV..WKROAD	Output	Working file

#### CREATION OF THE CONTINUATION EXTENSION DATA FILE: IDCAMS

This step executes a DELETE/DEFINE of the working file WYROAD

Code	Physical name	Type	Label
	&INDSV..WYROAD	Output	Working file

#### DATA RETRIEVAL: PTU200

This step selects 'data' type information in the initial sequential image and then formats the key of each record selected for the subsequent sort.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7BM	&&ROADBM	Input	User transactions
PAC7PC	&INDSN..BVPPE(0)	Input	Administration Database sequential image
PAC7PR	&&QS	Output	Formatted records (length=176 size = ND)
PAC7NX	&INDSV..WKROAD	Output	Long data
PAC7NY	&INDSV..WYROAD	Output	Unformatted data
PAC7AU	&&AU	Output	PR image (length=153)
PAC7YP	&&YP	Output	PY image (length=1,036)
PAC7EE		Report	Retrieval statistics output report

### SORT OF DATA: SORT

Sorting criteria: SRTREO1 member of the SY PDS.

Code	Physical name	Type	Label
SORTIN	&&QS	Input	Formatted records
SORTOUT	&&SQ	Output	Sorted records (length=176)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### PURGE: PTU210

This step reformats the records.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7PR	&&SQ	Input	Sorted records
PAC7PU	&&PU	Input	Entity records to be purged
PAC7BM	&&ROADBM	Input	User transactions
PAC7QS	&&QS	Output	Purged records (length=176, size= ND)
PAC7UM	&&UM	Output	Macro-structure call lines (length=176)

Code	Physical name	Type	Label
PAC7EE		Report	Library and session purge report
PAC7EK		Report	Entity-purge report
PAC7EB		Report	Technical report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0: OK
- 8: Capacity overload

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

#### INDEX REBUILDING: PTU220

This step rebuilds indexes using the data.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages file
PAC7BM	&&ROADBM	Input	User transactions
PAC7UR	&&QS	Input	Purged data
PAC7NX	&INDSV..WKROAD	Input	Long data
PAC7UM	&&UM	Input	Macro-structure call lines
PAC7PA	&&PA	Output	Data from frozen sessions (length=153 size=NH)
PAC7PB	&&PB	Output	Data from the current session (length=153 size=NC)
PAC7PC	&&PC	Output	First data record (length=153)
PAC7AN	&&AN	Output	Temporary index file (length=60 size=NI)
PAC7MR		Input/Output	Macro-structure call lines
PAC7EE		Report	Report of index building

#### SORT OF INDEXES: SORT

Sort criterion: SRTREO2 member of SY PDS.

Code	Physical name	Type	Label
SORTIN	&&AN	Input	Temporary index
SORTOUT	&&NA	Output	Sorted indexes
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### EXTENSION DATA PROCESSING: PTU226

Code	Physical name	Type	Label
PAC7NY	&INDSV..WYROAD	Input	Unformatted data
PAC7PA	&&PA	Input	Data from frozen sessions
PAC7PB	&&PB	Input	Data from the current session
PAC7PC	&&PC	Input	First data record
PAC7QA	&&QA	Output	Data from frozen sessions (length=153)
PAC7QB	&&QB	Output	Data from the current session (length=153)
PAC7QC	&&QC	Output	First data record (length=153)
PAC7QY	&&QY	Output	Long data (length=1018)

#### MERGE: PTU240

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

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7AN	&&NA	Input	Sorted index
PAC7AU	&&AU	Input	PR image
PAC7BM	&&ROADBM	Input	User transactions
PAC7PA	&&QA	Input	Data from frozen sessions
PAC7PB	&&QB	Input	Data from the current session
PAC7PC	&&QC	Input	First data record
PAC7QY	&&QY	Input	Extension data

Code	Physical name	Type	Label
PAC7CP	&INDSN..BVPPE(+1)	Output	Sequential image of the administration Database
PAC7IE		Report	Building the logical database

## ROAD - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - REORGANIZATION OF THE DATABASE ADMINISTRATOR -
/**
/** -----
/**
/** THE ROAD PROCEDURE ADMINI. MAY BE USED IN TWO CASES:
/** . WHEN PART OF THE DATA WAS DELETED BECAUSE OF A MAL-
/** FUNCTION 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 ENTITIES NOT
/** USED IN THE DATABASE ;
/**
/** -----
/**
/**BVPROAD  PROC INDSN='$INDSN',      INDEX OF NO VSAM SYSTEM FILES
/**          INDSV='$INDSV',          INDEX OF VSAM SYSTEM FILES
/**          OUT='$OUT',              OUTPUT CLASS
/**          CYL='(10,1)',            SORT WORKS SIZE
/**          SPAPE='(TRK,(100,10),RLSE)',  VA PAC DATABASE BACKUP
/**          STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
/**          SORTLIB='$BIBT',          SORT LIBRARY
/**          DSCB='$DSCB',            DSCB MODEL FILE
/**          UWK=$UWK,                WORK FILE UNIT
/**          VOLS='$SER=$VOLUN',       BACKUP FILE VOLUME
/**          UNITS=$UNITUN,           BACKUP FILE UNIT
/***:       SYSCAT='$SCAT',           VSAM SYSTEM CATALOG
/***:       VSAMCAT='$VCAT',         VSAM SYSTEM CATALOG
/**          PSBLIB='$PSBLIB',        LIBRARY OF PSB'S
/**          DBDLIB='$DBDLIB',        LIBRARY OF DBD'S
/**          RESLIB='$RESLIB',        IMS RESLIB
/**          PROCLIB='$PRCLIB',       IMS PROCLIB
/**          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
/**          CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$DBRC,IURL=$IURL
/** *****
/**INPUT    EXEC PGM=BVPTU001
/**STEPLIB  DD DSN=&STEPLIB,DISP=SHR
/**          DD DSN=$BCOB,DISP=SHR
/**CARTE    DD DDNAME=SYSIN
/**PAC7MB   DD DSN=&&ROADMB,DISP=(,PASS),
/**          UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
/**          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/** -----

```

```

//PTU2CL EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU2CL,PTU2CL$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//      &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7PC   DD DSN=&INDSN..BVP7PC(0),DISP=SHR
//PAC7EE   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&ROADMB,DISP=(OLD,PASS)
//PAC7BM   DD DSN=&&ROADBM,DISP=(NEW,PASS),
//      UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7PU   DD DSN=&&PU,DISP=(NEW,PASS),
//      UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//      DCB=(RECFM=FB,LRECL=50,BLKSIZE=4100)
//*-----
//DEFINE EXEC PGM=IDCAMS,COND=(0,NE,PTU2CL)
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DFWKROAD),DISP=SHR
//*-----
//DEFINE EXEC PGM=IDCAMS,COND=(0,NE,PTU2CL)
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DFWYROAD),DISP=SHR
//*-----
//PTU200 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU200,PTU200$SUG,&BUF,

```

```

//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//      &IRLM),
//      COND=(00,NE,PTU2CL)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//PAC7BM   DD DSN=&&ROADBM,DISP=(OLD,PASS)
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7EE   DD SYSOUT=&OUT
//PAC7NX   DD DSN=&INDSV..WKROAD,DISP=SHR
//PAC7NY   DD DSN=&INDSV..WYROAD,DISP=SHR
//PAC7PC   DD DSN=&INDSN..BVPPE(0),DISP=SHR
//PAC7PY   DD DUMMY
//PAC7AU   DD DSN=&&AU,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PR   DD DSN=&&QS,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//PAC7YP   DD DSN=&&YP,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=1036,BLKSIZE=27972)
//*-----
//SORTQS   EXEC PGM=SORT,COND=((0,NE,PTU2CL),(00,NE,PTU200))
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN   DD DSN=&&QS,DISP=(OLD,DELETE)
//SORTOUT  DD DSN=&&SQ,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//SYSIN    DD DSN=&INDSN..BVPSY(SRTRE01),DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//*-----
//PTU210   EXEC PGM=DFSRR00,REGION=$REGSIZ,

```

```

//      PARM=(DLI,BVPTU210,PTU210$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((00,NE,PTU2CL),(00,NE,PTU200),(0,NE, SORTQS))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7BM   DD DSN=&&ROADBM,DISP=(OLD,PASS)
//PAC7PR   DD DSN=&&SQ,DISP=(OLD,DELETE)
//PAC7PU   DD DSN=&&PU,DISP=(OLD,DELETE)
//PAC7QS   DD DSN=&&QS,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//PAC7UM   DD DSN=&&UM,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//PAC7EB   DD SYSOUT=&OUT
//PAC7EE   DD SYSOUT=&OUT
//PAC7EK   DD SYSOUT=&OUT
//*-----
//PTU220 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU220,PTU220$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((00,NE,PTU2CL),(00,NE,PTU200),(0,NE, SORTQS),
//      (00,NE,PTU210))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR

```



```

//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7EE DD SYSOUT=&OUT
//PAC7BM DD DSN=&&ROADBM,DISP=(OLD,PASS)
//PAC7UM DD DSN=&&UM,DISP=(OLD,PASS)
//PAC7UR DD DSN=&&QS,DISP=(OLD,PASS)
//PAC7AN DD DSN=&&AN,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=60,BLKSIZE=27960)
//PAC7MR DD UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=144,BLKSIZE=6336)
//PAC7PA DD DSN=&&PA,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PB DD DSN=&&PB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PC DD DSN=&&PC,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,1,RLSE),
//          DCB=(RECFM=FB,LRECL=153,BLKSIZE=153)
//PAC7NX DD DSN=&INDSV..WKROAD,DISP=SHR
//*-----
//SORTAN EXEC PGM=SORT,
// COND=((00,NE,PTU2CL),(0,NE,PTU200),(00,NE,SORTQS),
//       (00,NE,PTU210),(0,NE,PTU220))
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN DD DSN=&&AN,DISP=(OLD,PASS)
//SORTOUT DD DSN=&&NA,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=60,BLKSIZE=27960)
//SYSIN DD DSN=&INDSN..BVPSY(SRTRE02),DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//*-----
//PTU226 EXEC PGM=BVPTU226,COND=((0,NE,PTU2CL),(0,NE,PTU200),
// (0,NE,SORTQS),(0,NE,PTU210),(0,NE,PTU220),(0,NE,SORTAN))
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT

```

```

//SYSUDUMP DD SYSOUT=&OUT
//PAC7NY DD DSN=&INDSV..WYROAD,DISP=SHR
//PAC7PA DD DSN=&&PA,DISP=(OLD,DELETE)
//PAC7PB DD DSN=&&PB,DISP=(OLD,DELETE)
//PAC7PC DD DSN=&&PC,DISP=(OLD,DELETE)
//PAC7QA DD DSN=&&QA,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7QB DD DSN=&&QB,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7QC DD DSN=&&QC,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(TRK,1,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=153)
//PAC7QY DD DSN=&&QY,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=1018,BLKSIZE=27486)
//*-----
//PTU240 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU240,PTU240$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=((00,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
// (0,NE,PTU210),(0,NE,PTU220),(0,NE,SORTAN))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7IE DD SYSOUT=&OUT
//PAC7BM DD DSN=&&ROADBM,DISP=(OLD,DELETE)
//PAC7AU DD DSN=&&AU,DISP=(OLD,DELETE)
//PAC7AN DD DSN=&&NA,DISP=(OLD,DELETE)
//PAC7PA DD DSN=&&QA,DISP=(OLD,DELETE)
//PAC7PB DD DSN=&&QB,DISP=(OLD,DELETE)
//PAC7PC DD DSN=&&QC,DISP=(OLD,DELETE)
//PAC7QY DD DSN=&&QY,DISP=(OLD,DELETE)
//PAC7CP DD DSN=&INDSN..BVPPE(+1),DISP=(,CATLG,DELETE),
// UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPE,

```

```

//          DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//PAC7PD   DD DUMMY
//PAC7PY   DD DUMMY
//*-----
//DELETE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DLWKROAD),DISP=SHR
//DELETE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DLWYROAD),DISP=SHR
//

```

---

## RSAD - Restoration

### RSAD - Introduction

The purpose of this procedure is to rebuild the administration Database with the use of the sequential image resulting from the execution of the backup procedure (PACG).

It is also used to retrieve archived transactions via the resulting sequential image and to modify the number of 'gaps' of the Database.

To keep users and profiles updated since the last backup, the Administration database backup procedure must be first executed.

#### EXECUTION CONDITIONS

The administration Database must be closed to on-line use.

As the procedure rebuilds the Database, it may be interesting first to change the size of the Database files in expectation of an increase.

These changes must be done in the system parameters library.

The procedure re-initializes the transactions Journal physically and logically. It is therefore required to execute first the archiving procedure (ARAD) to backup the Journal.

#### ABNORMAL EXECUTION

Refer to subchapter 'Abnormal Endings' in chapter 'Overview'.

Whatever the reason of the abnormal ending, the procedure can be restarted once the problem solved.

## RSAD - Input / Processing / Results

A '\*' line with user code and password.

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

Position	Length	Value	Meaning
2	1	'Y'	Line code
3	5	nnnnn	Number of gaps as absolute value
8	2	pp	OR number of gaps as % (1)
10	2		Language code (FR, EN, ...)
12	1	'0'	No inhibition of the Journal
		'1'	Journal inhibition (no journalization of updated transactions)
		' '	Retrieval of the last value
14	3	'REC'	If archived transactions retrieved
26	4	'nnnn'	CHECKPOINT frequency (set to zero by default, only used for IMS , UNISYS, GCOS7, GCOS8, if REC in column 14)

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

**NOTES:** Where there is no input, the Database characteristics remain unchanged.

Any field left blank will be filled in with the current options.

The number of gaps entered is the minimum number for the Database. If the Database already contains more gaps than the number requested in 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.

If the Journal is inhibited ( parameter set to '1'), update transactions are not backed-up in the Journal file. In this case, it is impossible to restore the Database using the recovery of archived transactions ('REC' parameter on user input). It is therefore highly recommended to set this parameter to '0' (which is the default value) so as 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.

### PRINTED REPORT

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

### RESULTS

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

If the resulting Database contains no gaps, it is still possible to perform 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 that of the sequential image, or of the most recent transaction if the retrieval of archived transactions has been run.

## **RSAD - Description of Steps**

### NOTE

The &ARADJRNL parameter is by default set to NULLFILE in the installation. For a standard operating of the database, this parameter must contain the DSNAME of the archived transactions journal as following:  
ARADJRNL='\$INDSN..BVPJ(0)'

To retrieve archived transactions, position 'REC' on the input parameter line of the procedure.

### USER INPUT RECOGNITION: PTU010

Code	Physical name	Type	Label
CARTE		Input	User parameters
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data

Code	Physical name	Type	Label
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7PC	&INDSN..&BVPPE(0)	Input	Sequential image of the Administration Database
PAC7MB	&&RSADMB	Output	Parameter
PAC7DD		Report	Batch procedure authorization option

Return code:

- 8: No authorization for batch procedure

#### VALIDATION OF JOURNAL EXISTENCE: IDCAMS

LISTCAT on the Journal file (GJ).

Return code:

- 0 : The Journal file exists
- OTHER: The Journal file does not exist

#### VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed if the Journal file exists.

Code	Physical name	Type	Label
PAC7MB	&&RSADMB	Input	User transactions
BVP7AE	&INDSV..BVPAE	Input	Error messages file
BVP8GJ	&INDSV..BVPGJ	Input	Journal file
PAC7EU		Report	(only if the Journal was not archived)

Return code:

- 0: The Journal file was archived
- 8 : The Journal file was not archived (none of the RSAD steps was executed).

#### DEFINITION OF FILES: IDCAMS

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

Code	Physical name	Type	Label
	&INDSV..BVPGR	Output	Administration Database data
	&INDSV..BVPGN	Output	Administration Database index
	&INDSV..BVPGJ	Output	Administration Database Journal
	&INDSV..BVPGY	Output	Administration Database extension

#### RESTORATION OF THE DATABASE: PTU400

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

Code	Physical name	Type	Label
BVP7AE	&INDSV.&BVPAE	Input	Error labels
PAC7MB	&&RSADMB	Input	User transactions
PAC7PC	&INDSN..BVPPE(0)	Input	Sequential image of the administration Database
BVP8GR	&INDSV..BVPGR	Output	Administration Database data
BVP8GY	&INDSV..BVPGY	Output	Administration Database extension data
BVP8GN	&INDSV..BVPGN	Output	Administration Database index
BVP8GJ	&INDSV..BVPGJ	Output	Administration Database Journal
PAC7PS	&&PAC7PS	Output	Work file (2 records, length=144)
PAC7EU		Report	Restoration report
PAC7DD		Report	Batch procedure authorization option

#### DATABASE AVAILABILITY - TRANSACTIONS RETRIEVAL: PTU420

This step is executed if the Journal file has been archived. It updates the first record of the Data file.

Warning: this step is REQUIRED to obtain a consistent database.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7MB	&&RSADMB	Input	User transactions
BVP8GR	&INDSV..BVPGR	Input/Output	Administration Database data
PAC7JO	&ARADJRNL	Input	Journal to apply

Code	Physical name	Type	Label
PAC7PS	&&PAC7PS	Input	Work file
PAC7OJ	&&PAC7OJ	Output	Update transactions (length=170)
PAC7EU		Report	Retrieval report

Return codes:

- 0: There are transactions to retrieve
- 4: No transaction to retrieve  
OR erroneous user input.

In case of abnormal ending, the database cannot be updated.

#### UPDATE OF THE ADMINISTRATION DATABASE: PACA15

Code	Physical name	Type	Label
BVP8GR	&INDSV..BVPGR	Output	Administration Database data
BVP8GN	&INDSV..BVPGN	Output	Administration Database index
BVP8GY	&INDSV..BVPGY	Output	Administration Database extension
BVP8GJ	DUMMY	Output	Administration Database Journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC\$BASE	DUMMY	Input	DSMS file of the development Database
B7D3\$BASE	DUMMY	Input	DSMS index of the development Database
PAC7ME	DUMMY	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)



Code	Physical name	Type	Label
PAC7IF		Report	Erroneous transactions list (length=132)

The list of a user's transactions is preceded by a banner specifying the user's code.

Return codes :

- 0: OK, no error
- 2: warning
- 4: critical error

### RSAD - Execution JCL

```

/* -----
/*          VISUALAGE PACBASE
/*
/* -----
/*          - RELOADING RESTORATION OF THE DATABASE -
/*
/* -----
/*
/* INPUT
/* COL 2      : "Y"
/* COL 3-7    : NUMBER OF GAPS IN ABSOLUTE VALUE
/* COL 8-9    : NUMBER OF GAPS IN PERCENTAGE ( / BASE )
/* COL 10-11  : INITIAL LANGUAGE CODE (FR, EN, ..)
/* COL 12     : "1" INHIBITION OF TRANSACTION LOG
/* COL 14-16  : "REC" FOR RECOVERY OF ARCHIVED TRANSACTIONS
/* COL 26-29  : CKECKPOINT FREQUENCY
/*           : "D" : DISPATCH (3 FILES)
/*
/* IN THE ABSENCE OF INPUT, THE RELOAD DOES NOT MODIFY THE
/* NUMBER OF EXISTING GAPS, AND OTHER DATA IS UNCHANGED.
/*
/* IF THE JOURNAL FILE OF TRANSACTIONS ON DISK (GJ) IS NOT
/* REINITIALIZED, THE RESTORE CHAIN IS NOT EXECUTED.
/* IT IS THEREFORE NECESSARY TO EXECUTE THE ARCH PROCEDURE
/* FIRST.
/* -----
//BVPR SAD   PROC INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',             INDEX OF SYSTEM NO VSAM FILES
/*:         VSAMCAT='$VCAT',             USER VSAM CATALOG
/*:         SYSTCAT='$SCAT',            VA PAC SYSTEM VSAM CATALOG
//          OUT='$OUT',                 OUTPUT CLASS
//          VOLSV='$VOLSV',             VOLUME OF SYSTEM VSAM FILES
//          UNITS='$UNITSV',           UNIT OF SYSTEM VSAM FILES
//          ARADJRNL=NULLFILE,         DSN OF JOURNAL TO APPLY
//          SPAGJ='(TRK,(10,05),RLSE)', EXTRACTED JOURNAL SPACE
//          SPAGR='(CYL,(6,2),RLSE)',   SPACE 'GR' (OSAM DATABASE)

```

```

//          SPAGY='(CYL,(6,2),RLSE)', SPACE 'GY' (OSAM DATABASE)
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          PSBLIB='$PSBLIB', LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB', LIBRARY OF DBD'S
//          RESLIB='$RESLIB', IMS RESLIB
//          PROCLIB='$PRCLIB', IMS PROCLIB
//          UWK=$UWK, WORK UNIT
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU010,PTU010$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&RSADMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PAC7DD DD SYSOUT=&OUT
//PAC7PC DD DSN=&INDSN..BVP7PC(0),DISP=SHR
//PAC7PD DD DUMMY
//PAC7PY DD DUMMY
//*-----
//EXISAJ EXEC PGM=IDCAMS,COND=(00,NE,INPUT)
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(LIBVPGJ),DISP=SHR
//*-----
//PTU380 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU380,GTU380$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,

```

```

//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((00,NE,INPUT),(00,NE,EXISAJ))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GJ   DD DSN=&INDSV..BVP8GJ,DISP=SHR
//PAC7MB   DD DSN=&&RSADMB,DISP=(OLD,PASS)
//PAC7EU   DD SYSOUT=&OUT
//*-----
//DEFINE EXEC PGM=IDCAMS,COND=((0,NE,INPUT),(0,NE,PTU380))
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DFBVPGN),DISP=SHR
//*
//OSAMGJ1 EXEC PGM=IDCAMS,COND=((00,NE,INPUT),(00,NE,PTU380))
//SYSIN    DD DSN=&INDSN..BVPSY(DLBVPGJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAMGJ2 EXEC PGM=IEFBR14,COND=((00,NE,INPUT),(00,NE,PTU380))
//BVP8GJ   DD DSN=&INDSV..BVP8GJ,UNIT=&UNITS,
//      DISP=(,CATLG,DELETE),VOL=SER=&VOLS,
//      DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//      SPACE=&SPAGJ
//*
//OSAMGR1 EXEC PGM=IDCAMS,COND=((00,NE,INPUT),(00,NE,PTU380))
//SYSIN    DD DSN=&INDSN..BVPSY(DLBVPGR),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAMGR2 EXEC PGM=IEFBR14,COND=((00,NE,INPUT),(00,NE,PTU380))
//BVP8GR   DD DSN=&INDSV..BVP8GR,UNIT=&UNITS,
//      DISP=(,CATLG,DELETE),VOL=SER=&VOLS,
//      DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//      SPACE=&SPAGR
//*
//OSAMGY1 EXEC PGM=IDCAMS,COND=((00,NE,INPUT),(00,NE,PTU380))
//SYSIN    DD DSN=&INDSN..BVPSY(DLBVPGY),DISP=SHR

```

```

//SYSPRINT DD SYSOUT=&OUT
//*
//OSAMGY2 EXEC PGM=IEFBR14,COND=((00,NE,INPUT),(00,NE,PTU380))
//BVP8GY DD DSN=&INDSV..BVPGY,UNIT=&UNITS,
// DISP=(,CATLG,DELETE),VOL=SER=&VOLS,
// DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
// SPACE=&SPAGY
//*-----
//PTU400 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU400,GTU400$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=((00,NE,INPUT),(00,NE,PTU380),
// (00,NE,DEFINE),
// (00,NE,OSAMGJ2),(00,NE,OSAMGR2),(00,NE,OSAMGY2))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GJ DD DSN=&INDSV..BVP8GJ,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GY DD DSN=&INDSV..BVP8GY,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7EU DD SYSOUT=&OUT
//PAC7MB DD DSN=&&RSADMB,DISP=(OLD,PASS)
//PAC7PC DD DSN=&INDSN..BVP7PC(0),DISP=OLD
//PAC7PD DD DUMMY
//PAC7PY DD DUMMY
//PAC7PS DD DSN=&&PAC7PS,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=144,BLKSIZE=144),
// SPACE=(TRK,1)
//*-----
//PTU420 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU420,GTU420$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,

```

```

//      &IRLM),
//      COND=( (00,NE,INPUT), (00,NE,PTU380), (00,NE,PTU400),
//      (00,NE,DEFINE),
//      (00,NE,OSAMGJ2), (00,NE,OSAMGR2), (00,NE,OSAMGY2) )
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB   DD DSN=&&RSADMB,DISP=(OLD,DELETE)
//BVP7AE   DD DSN=&INDSV..BVP AE,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7J0   DD DSN=&ARADJRNL,DISP=OLD,DCB=BLKSIZE=27880
//PAC70J   DD DSN=&&PAC70J,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPAGJ,
//          DCB=(RECFM=FB,LRECL=170,BLKSIZE=27880)
//PAC7PS   DD DSN=&&PAC7PS,DISP=(OLD,PASS)
//PAC7EU   DD SYSOUT=&OUT
//*-----
//PACA15   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPACA15,GACA15$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM),
//          COND=( (00,NE,INPUT), (00,NE,PTU380), (00,NE,PTU400), (00,NE,PTU420),
//          (00,NE,DEFINE),
//          (00,NE,OSAMGJ2), (00,NE,OSAMGR2), (00,NE,OSAMGY2) )
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,

```

```

//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVP7GU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVP7GY,DISP=SHR
//BVP8GJ   DD DUMMY
//BVP8GN   DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GY   DD DSN=&INDSV..BVP8GY,DISP=SHR
//B7D3$BASE DD DUMMY
//B7DC$BASE DD DUMMY
//PAC7IE   DD SYSOUT=&OUT
//PAC7IF   DD SYSOUT=&OUT
//PAC7MV   DD DSN=&&PAC70J,DISP=(OLD,PASS)
//PAC7ME   DD DUMMY,DCB=BLKSIZE=353
//PAC7RB   DD DUMMY,DCB=BLKSIZE=80
//PAC7RY   DD DUMMY,DCB=BLKSIZE=310
//*
```

---

## Chapter 3. Database Management Utilities

---

### Backup Procedures

#### Introduction

This procedure is used to perform different types of operations on the development Database data according to the input code entered on the '\*' line.

#### PACS - Input Common to Managers

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	4	nnnn	Session number (UXSR-reserved) blank = current session
26	1		Session type (UXSR-reserved)
		'T'	If selection of frozen session
		' '	If selection of current session
29	4	cccc	Procedure function code (1)
33	1		Library extraction code (reserved to SASN)
		'A'	Extraction of a lower-level library and of its higher-level library
		'D'	Extraction of a library and of its dependent libraries
34	1		Library extraction option (reserved to SASN)
		'T'	Extraction of inter-library also (only if col 33 = 'D')
49	1		Locks extraction option (UXSR-reserved)
		' '	Locks extraction with user code = user code of '*' line
		'1'	No locks extraction
		'2'	Locks extraction with user code = source user code
67	1		(UXSR-reserved)

Position	Length	Value	Meaning
		'T'	If col 26 = '' selection of all the frozen sessions
		''	IF col 26 = '' selection of the current session only

(1) The different values of function codes are:

- MLIB: library management
- SAVE: development Database backup
- SASN: sub-networks backup
- UXSR: sub-networks extraction

## **Database Management**

### **MLIB - Introduction:**

The Database Management 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 procedure must be followed by the REST procedure so that the new library structure is taken into account.

### ABNORMAL EXECUTIONS

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

### **MLIB - Input / Processing / Results:**

There are two types of user input lines:

- Heading line (required) before all other input lines used to specify that a new VisualAge Pacbase Database is to be initialized or that an existing Database is to be modified.
- As many (optional) lines as there are Libraries to create, modify or delete.



The structure of the header is the following:

Position	Length	Value	Meaning
2	1	'G'	Line code
3	1	' '	Modification of existing Database
		'I'	Initialization of new Database
4	1	' '	Actual update
		'S'	Simulation

Update simulation is used to obtain the state of the Database as it would be if the requested modifications had actually been carried out.

It allows you to assess the impact of a change in the structure of the Database before actual execution. For large Databases, actual execution may use considerable machine time.

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

Position	Length	Value	Meaning
1	1	'C'	Creation
		'M'	Modification
		'A'	Deletion
		'I'	Clear the Library's content
2	1	'*'	Line code
3	3	bbb	Code of the Library to update
6	3	ccc	Code of the upper level Library
9	1	' '	Non initialized Library
		'V'	Virtual Library (only for the creation)

#### NOTE

Asterisks ("\*") are not authorized in Library codes.

#### UPDATE RULES

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

#### 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).

### CLEAR LIBRARY'S CONTENT TRANSACTION

Apply the same rules as for the deletion transaction.

### 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 be created by the same run of the procedure.

Once created, a Library has either a 'virtual' or 'not initialized' status.

- Virtual Libraries are created for future development projects. They are visible to the Administrator only. They cannot receive development specifications. You change their status to Not initialized in the Administrator workbench.
- Not initialized Libraries are visible to all users, but are not ready to receive development specifications. You change their status to Initialized in the Administrator workbench.

### WARNING

A VisualAge Pacbase Database cannot contain more than 300 Libraries.

### MODIFICATION TRANSACTIONS

Generally, transactions modify existing links between two Libraries. Usually a new Library is inserted between two existing Libraries.

When a new upper Library is assigned to a Library, the new Library must be empty and linked directly or indirectly to the former upper Library.

Structural errors are automatically detected.

It is not possible to cancel and recreate a Library during the same MLIB procedure.

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 after the update is printed.

### 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.

The execution report of a Database initialization does not include the Database name since it is later assigned via user input for the Restoration procedure.

### 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.

### NOTE

This procedure increments the session number.

## **Database Backup**

### **SAVE - Introduction:**

The Database Backup procedure (SAVE) saves the Database main files as sequential files: 'PC', 'PD' and 'PY' (logical names).

The backup is performed on the following files:

- Data file (AR),

- Index file (AN),
- Extension data file (AY).

An option allows for a backup of data files, indexes and extension data in three sequential files: ('AR' saved as 'PC', 'AN' saved as 'PD' and 'AY' as 'PY'). Otherwise, all three files are saved as one 'PC' file.

This option (Dispatch 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 closed.

#### ABNORMAL EXECUTION

Refer to chapter 'Overview', subchapter 'Abnormal Ending'.

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

The procedure can be restarted as it is once the problem has been solved.

#### EXECUTION OF ARCHIVING AND SAVE PROCEDURES

If the save procedure is preceded by a Journal archival (ARCH procedure), its execution may be conditioned by the return code of the PTU320 program of the ARCH procedure.

- 0: No error detected
- 8: Database not available

#### SIMPLIFIED BACKUP

Files may also be backed up via standard system utilities. In this case, run the SASY procedure to check the consistency of data and indexes (see Subchapter 'System Backup Complement').

#### PRINTED REPORTS

The procedure prints:

- a report containing the number of records saved in each file and the session number).

#### **SAVE - Input / Processing / Results:**

## OUTPUT

Depending on the 'Dispatch' option taken during restoration, the output of the backup procedure is the following:

- Either a single sequential file ('PC'), of variable length, containing the mirror of the three saved files,
- Or three sequential files ('PC', 'PD' and 'PY') of variable length.

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

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

## NOTE

This procedure increments the current session number.

### **Sub-Network Backup**

#### **SASN - Introduction:**

The Sub-Network Backup procedure (SASN) extracts one or more 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.

## NOTE

The MLIB procedure can produce a result similar to that produced by SASN. However, it keeps data 'gaps' in the backup and, unlike the SASN procedure, does not allow physical space to be gained.

## EXECUTION CONDITIONS

The database must be closed to on-line use.

## ABNORMAL EXECUTIONS

If an abend occurs, the procedure can be restarted as it is once the problem has been solved.

#### **SASN - User Input:**

The user must code one line per library to be extracted.

Position	Length	Value	Meaning
1	2	' '	
3	3	bbb	Code of lowest-level library of sub-network to be extracted. (All the upper-libraries of 'bbb' will be automatically extracted.)

## Partial Sub-Network Extraction

### UXSR - Introduction:

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

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

It is also possible to select:

- A frozen session (nT):

This frozen session will become the current session in the new Database. No other frozen session will be selected.

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

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

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

Examples:

- Creation of Libraries:

C\*CEN\_\_AAA (1)

C\*APPCENBBB (2)

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

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

- Merging of Libraries in the same Library:

C\*CEN\_\_CEN (1)

C\*APPCENAPP (2)

C\*APPCENBQQ (2)

- (1) Creation of the CEN Library with the contents of CEN.
- (2) Creation of the APP Library under the CEN Library with the contents of APP and BQQ.

The definition of the APP Library in the new Database will be identical to that of APP in the source Database since APP comes first, before BQQ.

- Renaming of Library:

C\*CEN\_\_AAA (1)

- (1) Creation of the CEN Library with the contents of AAA.

### CAUTION

No consistency checks are carried out; make sure you have entered valid user input lines.

It is not possible to copy an existing Library network and create new Libraries whose contents are identical to that of Libraries in the source network.

### EXECUTION CONDITIONS

On-line access must be closed.

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.

### **UXSR - User Input:**

There are two types of specific user input:

- Heading line (required) at the top of the input file that specifies if a simulation of the database is desired or not.
- As many lines (optional) as there are libraries to be created, modified or canceled.

The structure of the header is the following:

Position	Length	Value	Meaning
2	1	'G'	Line code
4	1	' '	Actual update
		'S'	Simulation

The simulation allows to obtain the state of the network after modifications, without actually carrying out these modifications. You can then assess the

impact of a modification in the network structure before carrying it out (it might prove to be costly in terms of machine time if the Database is very large).

You must enter as many (optional) lines as Libraries to be extracted for update.

Position	Length	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).

### **PACS - Description of Steps**

INPUT RECOGNITION: PTU001

FORMATTING SEQUENTIAL IMAGE :

IEBGENER

VSAM FILES CHECK: IDCAMS

FORMATTING SEQUENTIAL IMAGES: PTU520

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database extension data
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index



Code	Physical name	Type	Label
PAC7MB	&&PAC SMB	Input	Update transactions
PAC7PC	&NEWPC	Output	Database sequential image (length=1,023)
PAC7PD	&NEWPD	Output	If Dispatch option of backup: sequential image 2 of the database (length=1,023)
PAC7PY	&NEWPY	Output	If Dispatch option of backup: sequential image 3 of the database (length=1,023)
PAC7RP	&&PACSRP	Output	Sequential image of data (length=153) (should be able to contain all data)
PAC7NA	&&PAC SNA	Output	Sequential image of index (length=59) (should be able to contain all data)
PAC7NB	&&PACSNB	Output	Image of unsorted index (length=59)
PAC7RY	&&PAC SRY	Output	Sequential image of long data (length=1,019)
PAC7RQ	&&PACSPQ	Output	Temporary storage (1 record, length=153)
PAC7EV		Report	User transactions list
PAC7EU		Report	State of library before and after
PAC7EW		Report	Backup report
PAC7DD		Report	Abnormal endings report

Return codes:

- 2: MLIB or SASN and no error. Execution of PTU530.
- 4: MLIB and database simulation
- 8: Inconsistency of the database or no authorization for batch procedure

#### FORMATTING SEQUENTIAL IMAGES: PTU530

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV.&BASE.AR	Input	Development Database data
PAC7RP	&&PACSRP	Input	Data sequential image
PAC7NA	&&PAC SNA	Input	Index sequential image

Code	Physical name	Type	Label
PAC7NB	&&PACSNB	Input	unsorted index image
PAC7RY	&&PACSRY	Input	extension data sequential image
PAC7RQ	&&PACSRQ	Input	Intermediary storage
PAC7PC	&NEWPC	Output	Development Database sequential image
PAC7PD	&NEWPD	Output	If backup Dispatch option: sequential image 2 of the network
PAC7PY	&NEWPY	Output	If backup Dispatch option: sequential image 3 of the network
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Response to return code:

If the program sends a return code greater than '2', the backup is deleted by the next step in the procedure and a restoration must be performed using the last valid backup.

If there is no other backup, the user should first contact the product support to analyze the problem, then, the inconsistent database should be saved by the same procedure with the backup deletion step inactive. The resulting backup contains only data, and can only be used after running the REOR procedure.

#### BACKUP DELETION IN CASE OF DATABASE INCONSISTENCY: IEFBR14

Code	Physical name	Type	Label
DDPC	&INDUN..&BASE.PC(+1)	Input	Database sequential image
DDPD	&INDUN..&BASE.PD(+1)	Input	If backup Dispatch option: Database sequential image 2
DDPY	&INDUN..&BASE.PY(+1)	Input	If backup Dispatch option: Database sequential image 3

This step is executed if the return code of the previous program is greater than 2.

### **PACS - Execution JCL**

```

/** -----
/**      VISUALAGE PACBASE
/**

```

```

//* -----
//*          - BACKUP OF THE DATABASE -
//* -----
//*
//BVPPACS  PROC BASE=$BASE,      CODE OF VAPAC DATABASE
//          INDSN='$INDSN',      INDEX OF SYSTEM NO VSAM FILES
//          INDSV='$INDSV',      INDEX OF SYSTEM VSAM FILES
//          INDUV='$INDUV',      INDEX OF USER VSAM FILES
//*:       SYSTCAT='$VCAT',      VSAM SYSTEM CATALOG
//*:       VSAMCAT='$SCAT',      VSAM USER CATALOG
//          DSCB='$DSCB',        DSCB MODEL FILE
//          OUT='$OUT',          OUTPUT CLASS
//          NEWPC=$INDUN..$BASE.PC(+1),  SAVE FILE PC
//          NEWPD=$INDUN..$BASE.PD(+1),  SAVE FILE PD
//          NEWPY=$INDUN..$BASE.PY(+1),  SAVE FILE PY
//          VOLS='SER=$VOLUN',    DATABASE BACKUP VOLUME
//          UNITS='$UNITUN',      BACKUP UNIT (DISK OR CARTRIDGE)
//          VOLS2='SER=$VOLUN',    DATABASE 2ND BACKUP VOLUME
//          UNITS2='$UNITUN',    2ND BACKUP UNIT (DISK OR CARTRIDGE)
//          VOLS3='SER=$VOLUN',    DATABASE 3RD BACKUP VOLUME
//          UNITS3='$UNITUN',    3RD BACKUP UNIT (DISK OR CARTRIDGE)
//          SPAPC=(TRK,(300,10)),    VA PAC DATABASE BACKUP
//          SPAPD=(TRK,(300,10)),    VA PAC DATABASE BACKUP 2
//          SPAPY=(TRK,(300,10)),    VA PAC DATABASE BACKUP 3
//          SPAAN=(TRK,(700,10),RLSE)', SPACE TEMPORARY INDSV
//          SPAAR=(TRK,(700,10),RLSE)', SPACE TEMPORARY DATA
//          CYL=(10,1)',            TEMPORARY SPACE
//          SORTLIB='$BIBT',      SORT LIBRARY
//          UWK='$UWK',           WORK UNIT
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULE
//          PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//          RESLIB='$RESLIB',     IMS RESLIB
//          PROCLIB='$PRCLIB',    IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT    EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//PAC7MB  DD DSN=&&PAC5MB,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(3,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//CARTE   DD DDNAME=SYSIN
//IEBGN1  EXEC PGM=IEBGENER
//*-----
//SYSPRINT DD DUMMY
//SYSIN   DD DUMMY
//SYSUT1  DD DUMMY
//SYSUT2  DD DSN=&NEWPC,
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,
//          VOL=&VOLS,

```

```

//          SPACE=&SPAPC,
//          DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//IEBGN2 EXEC PGM=IEBGENER
//*-----
//SYSPRINT DD DUMMY
//SYSIN    DD DUMMY
//SYSUT1   DD DUMMY
//SYSUT2   DD DSN=&NEWPD,
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS2,
//          VOL=&VOLS2,
//          SPACE=&SPAPD,
//          DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//IEBGN3 EXEC PGM=IEBGENER
//*-----
//SYSPRINT DD DUMMY
//SYSIN    DD DUMMY
//SYSUT1   DD DUMMY
//SYSUT2   DD DSN=&NEWPY,
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS3,
//          VOL=&VOLS3,
//          SPACE=&SPAPY,
//          DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//*-----
//PTU520 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU520,PTU520$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT

```

```

//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7MB DD DSN=&&PAC7MB,DISP=(OLD,PASS)
//PAC7PC DD DSN=&NEWPC,DISP=(OLD,,DELETE)
//PAC7PD DD DSN=&NEWPD,DISP=(OLD,,DELETE)
//PAC7PY DD DSN=&NEWPY,DISP=(OLD,,DELETE)
//PAC7NA DD DSN=&&PAC7NA,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=&SPAAN,
// DCB=(RECFM=FB,LRECL=59,BLKSIZE=27966)
//PAC7NB DD DSN=&&PAC7NB,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=&SPAAN,
// DCB=(RECFM=FB,LRECL=59,BLKSIZE=27966)
//PAC7RP DD DSN=&&PAC7RP,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=&SPAAR,
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7RQ DD DSN=&&PAC7RQ,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=(TRK,1,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=153)
//PAC7RY DD DSN=&&PAC7RY,DISP=(,PASS,DELETE),
// UNIT=&UWK,SPACE=&SPAPY,
// DCB=(RECFM=FB,LRECL=1019,BLKSIZE=27513)
//PAC7EU DD SYSOUT=&OUT
//PAC7EW DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7EV DD SYSOUT=&OUT
//*-----
//PTU530 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU530,PTU530$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM),
// COND=(2,NE,PTU520)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR

```

```

//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7RQ DD DSN=&&PACSRQ,DISP=(OLD,DELETE)
//PAC7RP DD DSN=&&PACSRP,DISP=(OLD,DELETE)
//PAC7RY DD DSN=&&PACSRY,DISP=(OLD,DELETE)
//PAC7NA DD DSN=&&PAC7NA,DISP=(OLD,DELETE)
//PAC7NB DD DSN=&&PAC7NB,DISP=(OLD,DELETE)
//PAC7PC DD DSN=&NEWPC,DISP=(OLD,,DELETE)
//PAC7PD DD DSN=&NEWPD,DISP=(OLD,,DELETE)
//PAC7PY DD DSN=&NEWPY,DISP=(OLD,,DELETE)
//*-----
//DELPC EXEC PGM=IEFBR14,COND=((2,GT,PTU520),(2,GT,PTU530))
//*-----
//DDPC DD DSN=&NEWPC,DISP=(OLD,DELETE)
//DDPD DD DSN=&NEWPD,DISP=(OLD,DELETE)
//DDPY DD DSN=&NEWPY,DISP=(OLD,DELETE)

```

---

## UPDT - Freeze

### UPDT - Introduction

The Database Update procedure (UPDT) is used to freeze the database and executes also a Batch update of the database.

The principle of sessions makes it possible to manage various versions of the same application. The administrator then freezes the Database, creating a snapshot of the current session.

The procedure allows access to all libraries which make up the database, according to the different user authorizations.

The update can be executed in on-line mode.

For further information about UPDT, refer to 'the Developer's Procedures' manual.

## UPDT - Input

A '\*' line with the user code and password.

No other update should precede the 'X1HIST' line.

Position	Length	Value	Meaning
2	6	'X1HIST'	Line code for a session freeze
8	50		Session Name
58	4		Session number
65	1		Session status
		' '	Frozen and updateable session
		'N'	Frozen and non-updateable session
		'D'	Deleted session
66	15		Session Short Name

## UPDT - Description of Steps

INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

TRANSACTIONS FORMATTING: PACA05

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database Data file
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database Index file
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database extension data
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database Index file
BVP7GR	&INDSV..BVP GR	Input	Administration Database Data file
BVP7GU	&INDSV..BVP GU	Input	Administration Database Users
PAC7MB	&&UPDTMB	Input	Update transactions
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MV	&&PAC7MV	Output	Formatted transactions (length=170, must be able to contain all input transactions plus the elementary delete transactions generated by the multiple delete transactions)

Code	Physical name	Type	Label
PAC7MW		Output	Work file

#### UPDATE OF THE DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV.&BASE.AR	Output	Development Database Data file
B7AN\$BASE	&INDUV.&BASE.AN	Output	Development Database index
B7AY\$BASE	&INDUV.&BASE.AY	Output	Development Database extension
B7AJ\$BASE	&INDUV.&BASE.AJ	Output	Development Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC\$BASE	&DSMS.&BASE.DC	Input	DSMS file of Development Database Elements
B7D3\$BASE	&DSMS.&BASE.D3	Input	Index of Development Database Elements
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	&&PAC7RB	Output	UPDT erroneous transactions (length=80)
PAC7RY	&&PAC7RY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

The list of user transactions is preceded by a banner with the user code.

Return codes:

- 0: OK, no error
- 2: Warning



- 4: Critical error

## UPDT - Execution JCL

```

//* -----
//*          VISUALAGE PACBASE
//*
//* -----
//*          - BATCH UPDATE -
//*
//* -----
//*
//* REFER TO THE BATCH FORMS AND TO THE DESCRIPTION OF THE
//* INPUT CORRESPONDING TO EACH ENTITY.
//*
//* INPUT :
//* - USER IDENTIFICATION LINE (REQUIRED)
//*   COL 2 : "*"
//*   COL 3 : USERIDXX
//*   COL 11 : PASSWORD
//*   COL 28 : LANGUAGE CODE, USEFUL WHEN TRANSACTION ARE
//*           NOT IN THE SAME LANGUAGE AS THE DATABASE.
//*   COL 67 : "N" NOT UPPERCASE/LOWERCASE CONVERSION
//* - COMMAND LINE
//*   THE LIST OF ALL AVAILABLE VALUES FOR THE ENTITY
//*   TO BE UPDATED IS FOUND IN REFERENCE MANUAL.
//*
//* -----
//*
//BVPUPD   PROC BASE=$BASE,          CODE OF VAPAC DATABASE
//         INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NO VSAM FILES
//         INDUV='$INDUV',          INDEX OF USER VSAM FILES
//         DSMS='$DSMS',            INDEX OF PRODUCT ELEMENTS DSM
//*:       VSAMCAT='$VCAT',          VSAM USER CATALOG
//*:       SYSCAT='$SCAT',          VSAM SYSTEM CATALOG
//         OUT='$OUT',              OUTPUT CLASS
//         OUTL='$OUTL',            OUTPUT CLASS
//         SPAMB='(TRK,(100,10),RLSE)', SPACE OF TRANSACTIONS
//         SPALG='(CYL,(5,2),RLSE)',  IMS LOG SPACE FOR ROLL
//         STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//         PSBLIB='$PSBLIB',         LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',         LIBRARY OF DBD'S
//         RESLIB='$RESLIB',         IMS RESLIB
//         PROCLIB='$PRCLIB',        IMS PROCLIB
//         UWK=$UWK,                 WORK UNIT
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$DBRC,IRLM=$IRLM,BKO=Y
//*****
//INPUT    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&UPDTMB,DISP=(,PASS),UNIT=&UWK,
//         SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//CARTE    DD DDNAME=SYSIN

```

```

//*-----
//VERIFY EXEC PGM=IDCAM5
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGY DD DSN=&INDSV..BVPGY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//*-----
//PACA05 EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,BVPACA05,PACA05$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,
// &DBRC,&IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR

```

```

//PAC7MB DD DSN=&&UPDTMB,DISP=(OLD,DELETE)
//PAC7ME DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=372,BLKSIZE=5952)
//PAC7MV DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW DD DISP=(,DELETE),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//*-----
//PACA15 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPACA15,PACA15$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM,,&BKO)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DSN=&&IMSLG,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPALG,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//BVP7GY DD DSN=&INDSV..BVP7GY,DISP=SHR
//B7D3$BASE DD DSN=&DSMS..&BASE.D3,DISP=SHR
//B7DC$BASE DD DSN=&DSMS..&BASE.DC,DISP=SHR
//PAC7IE DD SYSOUT=&OUTL
//PAC7IF DD SYSOUT=&OUTL
//PAC7MV DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7ME DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7RB DD DSN=&&PAC7RB,DISP=(,PASS),UNIT=&UWK,
//      SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)

```

```
//PAC7RY DD DSN=&&PAC7RY,DISP=(,PASS),UNIT=&UWK,  
//          SPACE=&SPAMB,  
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200)  
//*  
//
```

---

## SASY - Database System Backup Complement

### SASY - Introduction

This Database Backup procedure saves the Database using any utility of the Operating System, while at the same time creating a checkpoint, through the incrementation of the session number.

The following files are to saved:

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

### EXECUTION CONDITIONS

The Data (AR) and Index (AN) files must have been saved.

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.

### ABNORMAL EXECUTION

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

The procedure can be restarted as it is once the problem has been solved.

### USER INPUT

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

### RESULT

This procedure increments the current session number.

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

## SASY - Description of Steps

INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

SESSION NUMBER INCREMENTATION: PTU502

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV.&BASE.AR	Input/Output	Development Database data
BVP7AE	&INDSV.&BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
PAC7MB	&&SASYMB	User transactions	User transactions
PAC7GZ		Report	Report
PAC7DD		Report	Batch procedure authorization option
PAC7DS		Report	Database validity report

## SASY - Execution JCL

```
/** -----  
/**      VISUALAGE PACBASE  
/**  
/** -----  
/**      - COMPLEMENT OF "SYSTEM" BACKUP OF THE DATABASE-  
/**  
/** -----  
/**  
/** THE DATABASE SYSTEM BACKUP COMPLEMENT PROCEDURE  
/** (SASY) ALLOWS YOU TO SAVE THE DATABASE USING ANY  
/** UTILITY OF THE OPERATING SYSTEM, WHILE AT THE  
/** SAME TIME CREATING A CHECKPOINT, THROUGH THE  
/** INCREMENTATION OF THE SESSION NUMBER.  
/** THE FOLLOWING FILES ARE TO BE BACKED UP:  
/**   . DATA FILE (AR),  
/**   . INDEX FILE (AN).  
/**  
/** -----  
/**  
/** BVPSASY PROC BASE=$BASE,      CODE OF VAPAC DATABASE  
/**      INDSN='$INDSN',          INDEX OF NON VSAM SYSTEM FILES  
/**      INDUV='$INDUV',          INDEX OF VSAM USER FILES  
/**      INDSV='$INDSV',          INDEX OF VSAM SYSTEM FILES  
/***:      VSAMCAT='$VCAT',        USER VSAM CATALOG  
/***:      SYSCAT='$SCAT',         VA PAC SYSTEM VSAM CATALOG
```

```

// STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
// OUT='$OUT', OUTPUT CLASS
// UWK=$UWK, WORK UNIT
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****-----*
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&SASYMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-----*
//VERIFY EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----*
//PTU502 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU502,PTU502$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY

```

```

//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB DD DSN=&&SASYMB,DISP=(OLD,PASS)
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7GZ DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7DS DD SYSOUT=&OUT
//

```

---

## REST - Database Restoration

### REST - Introduction

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

It also allows the retrieval of archived transactions after this sequential image has been produced 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.

#### ABNORMAL EXECUTIONS

Refer to chapter 'Overview', subchapter 'Abnormal Ending'.

If an abend occurs, the procedure can be restarted as it is once the problem has been solved.

#### CHECKPOINT REQUEST

This facility allows you to request synchronization points during a batch update (UPDT procedure) or during a database restoration (REST or RESY procedures).

In case of ABEND, a ROLLBACK is performed, thus securing a coherent database.

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

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

#### INPUT OF THE CHECKPOINT FREQUENCY RATE FOR A BATCH UPDATE

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

Position	Length	Value	Meaning
2	1	Y	Line code
4	4	nnnn	Checkpoint frequency rate
			(Default value=0000)

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

### **REST - Input / Processing / Results**

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

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

Position	Length	Value	Meaning
2	1	Y	Line code
3	5	nnnnn	Number of gaps
8	2	pp	Number of gaps as a percentage
10	2		Language code (EN, FR, ...)
12	1	0	No inhibition of journal
		1	Inhibition of journal (no
			journalization of update transactions)
		' '	Retrieval of previous value
14	3	REC	If archived transactions recovered



Position	Length	Value	Meaning
17	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
21	4	nnnn	Maximum number of accesses for on-line searches in the Database (default value: 300)
25	1	U	Implicit update (default option)
		N	Explicit update
26	4	nnnn	Checkpoint frequency (IMS, UNISYS, GCOS7, and GCOS8 only) if REC in col. 13 (default: nnnn=0000)
36	12		PFkeys assigned functions (2).
79	1		Dispatch option of Backup:
		'D'	Dispatch: sequential backup of the database in two separate files.
		'N'	No Dispatch: standard backup of the database in one PC file.
		' '	Retrieval of previous value

(1) The user can insert 'gaps' into the database via the system corresponding sub-menu which will be used in case of new data creation in the database.

(2): PFKeys assignment:

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.

#### NOTE

Where there is no input, the Database characteristics remain unchanged.

Any field left blank will retrieve the current options.

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 to 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 line). It is therefore highly recommended to set this parameter to 0 (which is the default option), in order to avoid restoration problems.

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

### SIMPLIFIED RESTORATION

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

### PRINTED REPORT

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

### 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 perform 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.

### **REST - Description of Steps**

#### USER INPUT RECOGNITION: PTU010

Code	Physical name	Type	Label
CARTE		Input	User parameters
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the development Database
PAC7PD	&INDUN..&BASE.PD(0)	Input	If backup option Dispatch: sequential image 2 of the database
PAC7PY	&INDUN..&BASE.PY(0)	Input	If Dispatch option of the backup: sequential image 3 of the database
PAC7MB	&&RESTMB	Output	Parameter
PAC7DD		Report	Batch procedures authorization option

Return code:

- 8: no authorization for batch procedure

#### VALIDATION OF JOURNAL EXISTENCE: IDCAMS

LISTCAT of the journal file (AJ).

Return codes:

.0 : The Journal file exists

.OTHER: The Journal file does not exist

### VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed only if the Journal file exists.

Code	Physical name	Type	Label
PAC7MB	&&RESTMB	Input	User transactions
BVP7AE	&INDSV..BVPAE	Input	Error messages file
B7AJ\$BASE	&INDUV..&BASE.AJ	Input	Journal file
PAC7EU		Report	(only if the journal was not saved)

Return codes:

0: The Journal file was archived.

8: The Journal file was not archived (no REST step is executed).

### DEFINITION OF FILES: IDCAMS

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

Code	Physical name	Type	Label
	&INDUV..&BASE.AR	Output	Development Database data
	&INDUV..&BASE.AN	Output	Development database index
	&INDUV..&BASE.AJ	Output	Development Database Journal
	&INDUV..&BASE.AY	Output	Development Database extension

### RESTORATION OF THE DATABASE: PTU400

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

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7MB	&&RESTMB	Input	User transaction
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the development Database

Code	Physical name	Type	Label
PAC7PD	&INDUN..&BASE.PD(0)	Input	If backup option Dispatch: sequential image 2 of the database
PAC7PY	&INDUN..&BASE.PY(0)	Input	If Dispatch option in save procedure: sequential image 3 of the database
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN\$BASE	&INDUV..&BASE.AN	Output	Development Database index
B7AY\$BASE	&INDUV..&BASE.AY	Output	Development Database extension data
B7AJ\$BASE	&INDUV..&BASE.AJ	Output	Development Database Journal
PAC7PS	&&PAC7PS	Output	Work file (2 records, length=144)
PAC7EU		Report	Backup report

#### DATABASE AVAILABILITY - TRANSACTION RETRIEVAL: PTU420

This step is executed if the Journal file has been archived. It updates the first record of the Data file.

This step is REQUIRED to obtain a consistent database.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP7AE	Input	Error messages
PAC7MB	&&RESTMB	Input	User transactions
B7AR\$BASE	&INDUV..&BASE.AR	Input/Output	Development Database data
PAC7JO	&ARCHJRNL	Input	Journal to apply
PAC7PS	&&PAC7PS	Input	Work file
PAC7OJ	&&PAC7OJ	Output	Update transactions (length=170)
PAC7EU		Report	Retrieval report

Return codes:

- 0: Transactions to retrieve
- 4: No transactions to retrieve or error on user input.

In case of abend, the update cannot be performed.

## UPDATE OF THE DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV..&BASE.AR	Output	Development Database data
B7AN\$BASE	&INDUV..&BASE.AN	Output	Development Database index
B7AY\$BASE	&INDUV..&BASE.AY	Output	Development Database extension
B7AJ\$BASE	DUMMY	Output	Development Database Journal
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GY	&INDSV..BVP GY	Input	Administration Database extension
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
B7DC\$BASE	&DSMS..&BASE.DC	Input	DSMS file of Development Database Elements
B7D3\$BASE	&DSMS..&BASE.D3	Input	DSMS index of Development Database Elements
PAC7ME	DUMMY	Input	Work file
PAC7MV	&&PAC7OJ	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

- 0: OK, no error
- 2: Warning
- 4: Critical error

### REST - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**

```

```

/** -----
/**          - RELOADING RESTORATION OF THE DATABASE -
/** -----
/**
/** INPUT
/** COL 2      : "Y"
/** COL 3-7    : NUMBER OF GAPS IN ABSOLUTE VALUE
/** COL 8-9    : NUMBER OF GAPS IN PERCENTAGE ( / BASE )
/** COL 10-11  : INITIAL LANGUAGE CODE (FR, EN, ..)
/** COL 12     : "1" INHIBITION OF TRANSACTION LOG
/** COL 14-16  : "REC" FOR RECOVERY OF ARCHIVED TRANSACTIONS
/** COL 17-20  : 4 CHARACTERS TO BE DISPLAYED
/**           : ON ALL SCREEN OF THE PRODUCT
/** COL 21-24  : "NNNN" MAXIMUM NUMBER OF SEARCH ACCESSES
/**           : TO THE DATABASE(LISTS)-(DEFAULT VALUE:300)
/** COL 25     : "U" (DEFAULT VALUE) : IMPLICIT UPDATE
/**           : "N" EXPLICIT UPDATE
/** COL 26-29  : CKECKPOINT FREQUENCY
/** COL 36-47  : PF-KEYS SIGNIFICATIONS
/** COL 79     : BACKUP FILES DISPATCH
/**           : "N" (DEFAULT VALUE) : NO DISPATCH (1 FILE)
/**           : "D" : DISPATCH (3 FILES)
/**
/** IN THE ABSENCE OF INPUT, THE RELOAD DOES NOT MODIFY THE
/** NUMBER OF EXISTING GAPS, AND OTHER DATA IS UNCHANGED.
/**
/** IF THE JOURNAL FILE OF TRANSACTIONS ON DISK (AJ) IS NOT
/** REINITIALIZED, THE RESTORE CHAIN IS NOT EXECUTED.
/** IT IS THEREFORE NECESSARY TO EXECUTE THE ARCH PROCEDURE
/** FIRST.
/** -----
/**BVPREST  PROC BASE=$BASE,          CODE OF VAPAC DATABASE
/**          INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
/**          INDUV='$INDUV',          INDEX OF USER VSAM FILES
/**          INDUN='$INDUN',          INDEX OF USER NO VSAM FILES
/**          INDSN='$INDSN',          INDEX OF SYSTEM NO VSAM FILES
/**          DSMS='$DSMS',            INDEX OF DSMS FILES
/**          VOLU='$VOLUV',            VOLUME OF USER VSAM FILES
/**          UNITU='$UNITUV',          UNIT OF USER VSAM FILES
/***:       VSAMCAT='$VCAT',          VSAM USER CATALOG
/***:       SYSTCAT='$SCAT',          VSAM SYSTEM CATLOG
/**          OUT='$OUT',              OUTPUT CLASS
/**          ARCHJRNL=NULLFILE,       DSN OF JOURNAL TO APPLY
/**          SPAAJ='(CYL,(3,2),RLSE)', AJ FILE SPACE (OSAM DB)
/**          SPAAR='(CYL,(6,2),RLSE)', SPACE 'AR' (OSAM DATABASE)
/**          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
/**          PSBLIB='$PSBLIB',        LIBRARY OF PSB'S
/**          DBDLIB='$DBDLIB',        LIBRARY OF DBD'S
/**          RESLIB='$RESLIB',        IMS RESLIB
/**          PROCLIB='$PRCLIB',       IMS PROCLIB
/**          UWK=$UWK,                WORK UNIT
/**          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
/**          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
/*******

```

```

//INPUT EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU010,PTU010$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP GU,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&RESTMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PAC7DD DD SYSOUT=&OUT
//PAC7PC DD DSN=&INDUN..&BASE.PC(0),DISP=OLD
//PAC7PD DD DSN=&INDUN..&BASE.PD(0),DISP=SHR
//PAC7PY DD DSN=&INDUN..&BASE.PY(0),DISP=SHR
//*-----
//EXISAJ EXEC PGM=IDCAMS,COND=(0,NE,INPUT)
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(LI&BASE.AJ),DISP=SHR
//*-----
//PTU380 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU380,PTU380$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=((00,NE,INPUT),(00,NE,EXISAJ))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR

```



```

//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7MB DD DSN=&&RESTMB,DISP=(OLD,PASS)
//PAC7EU DD SYSOUT=&OUT
//*-----
//DEFINE EXEC PGM=IDCAMS,COND=((0,NE,INPUT),(0,NE,PTU380))
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DF&BASE.AN),DISP=SHR
//*
//OSAMAJ1 EXEC PGM=IDCAMS,COND=((00,NE,INPUT),(00,NE,PTU380))
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.AJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAMAJ2 EXEC PGM=IEFBR14,COND=((00,NE,INPUT),(00,NE,PTU380))
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,UNIT=&UNITU,
//      DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
//      DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//      SPACE=&SPAAJ
//*
//OSAMAR1 EXEC PGM=IDCAMS,COND=((00,NE,INPUT),(00,NE,PTU380))
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.AR),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAMAR2 EXEC PGM=IEFBR14,COND=((00,NE,INPUT),(00,NE,PTU380))
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,UNIT=&UNITU,
//      DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
//      DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//      SPACE=&SPAAR
//*
//OSAMAY1 EXEC PGM=IDCAMS,COND=((00,NE,INPUT),(00,NE,PTU380))
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.AY),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAMAY2 EXEC PGM=IEFBR14,COND=((00,NE,INPUT),(00,NE,PTU380))
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,UNIT=&UNITU,
//      DISP=(,CATLG,DELETE),VOL=SER=&VOLU,
//      DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//      SPACE=&SPAAJ
//*-----
//PTU400 EXEC PGM=DFSRR00,REGION=$REGSIZ,

```

```

//      PARM=(DLI,BVPTU400,PTU400$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((00,NE,INPUT),(00,NE,PTU380),(00,NE,DEFINE),
//      (00,NE,OSAMAJ2),(00,NE,OSAMAR2),(00,NE,OSAMAY2))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP AE,DISP=SHR
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&RESTMB,DISP=(OLD,PASS)
//PAC7PC   DD DSN=&INDUN..&BASE.PC(0),DISP=OLD
//PAC7PD   DD DSN=&INDUN..&BASE.PD(0),DISP=SHR
//PAC7PY   DD DSN=&INDUN..&BASE.PY(0),DISP=SHR
//PAC7PS   DD DSN=&&PAC7PS,DISP=(,PASS),UNIT=&UWK,
//      DCB=(RECFM=FB,LRECL=144,BLKSIZE=144),
//      SPACE=(TRK,1)
//*-----
//PTU420 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU420,PTU420$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((00,NE,INPUT),(00,NE,PTU380),(00,NE,PTU400),
//      (00,NE,DEFINE),
//      (00,NE,OSAMAJ2),(00,NE,OSAMAR2),(00,NE,OSAMAY2))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR

```

```

//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB DD DSN=&&RESTMB,DISP=(OLD,DELETE)
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7JO DD DSN=&ARCHJRNL,DISP=OLD,DCB=BLKSIZE=27880
//PAC7OJ DD DSN=&&PAC7OJ,DISP=(,PASS),
//      UNIT=&UWK,SPACE=&SPAAJ,
//      DCB=(RECFM=FB,LRECL=170,BLKSIZE=27880)
//PAC7PS DD DSN=&&PAC7PS,DISP=(OLD,PASS)
//PAC7EU DD SYSOUT=&OUT
//*-----
//PACA15 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPACA15,PACA15$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//      &IRLM),
//      COND=((00,NE,INPUT),(00,NE,PTU380),(00,NE,PTU400),(00,NE,PTU420),
//      (00,NE,DEFINE),
//      (00,NE,OSAMAJ2),(00,NE,OSAMAR2),(00,NE,OSAMAY2))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR

```

```

//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY DD DSN=&INDSV..BVPGY,DISP=SHR
//B7AJ$BASE DD DUMMY
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//B7D3$BASE DD DSN=&DSMS..&BASE.D3,DISP=SHR
//B7DC$BASE DD DSN=&DSMS..&BASE.DC,DISP=SHR
//PAC7IE DD SYSOUT=&OUT
//PAC7IF DD SYSOUT=&OUT
//PAC7MV DD DSN=&&PAC70J,DISP=(OLD,DELETE)
//PAC7ME DD DUMMY,DCB=BLKSIZE=353
//PAC7RB DD DUMMY,DCB=BLKSIZE=80
//PAC7RY DD DUMMY,DCB=BLKSIZE=310
//*
```

---

## RESY - Database System Restoration Complement

### RESY - Introduction

This procedure restores a Database that can be handled in on-line mode, from a System backup obtained through a utility followed by the SASY procedure.

It is executed after a System restoration utility to complete the restoration of the Data (AR), Index (AN), and Extension (AY) 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 line.

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, AR and AY files by the on-site system utility.

On-line access must be closed.

### ABNORMAL EXECUTIONS

If an abend occurs, the procedure can be restarted as it is once the problem has been solved.

### PRINTED REPORT

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 memorized in the new database.

## RESULTS

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

**NOTE:** After the procedure execution, the current session number is the session number of the restored image, or of the most recent transaction if the retrieval of archived transactions has been requested.

## CHECKPOINT REQUEST

This facility allows you to request synchronization points during a batch update (UPDT procedure) or during a database restoration (REST or RESY procedures).

In case of ABEND, a ROLLBACK is performed, thus securing a coherent database.

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

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

## INPUT OF THE CHECKPOINT FREQUENCY RATE FOR A BATCH UPDATE

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

Position	Length	Value	Meaning
2	1	Y	Line code
4	4	nnnn	Checkpoint frequency rate
			(Default value=0000)

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

### **RESY - Input / Processing / Results**

A '\*' line with user code and password.

The input has the following structure:

Position	Length	Value	Meaning
2	1	'Y'	Line code
10	1		Language code (FR, EN...)
12		'0'	No journal inhibition
		'1'	Inhibition of journal (update transactions are not journalized)
		' '	Retrieval of the last value
14	3	REC	if archived transactions are recovered
17	4	XXXX	4-character Database code chosen by the Database Manager (displayed in the top-right corner of all screens)
21	4	'nnnn'	Maximum number of accesses for on-line searches in the Database (default value: 300)
25	1	U	Implicit update (default option)
		N	Explicit update
26	4	'nnnn'	Checkpoint frequency rate (IMS, UNISYS, GCOS7, and GCOS8 only) if REC in col. 13 (default: nnnn=0000)
36	12		PFkeys assigned functions (1)
79	1		Dispatch option of backup:
		'D'	Dispatch Sequential backup of the database on two separate files.
		'N'	No Dispatch Standard backup on a single PC file.
		' '	Same as previous execution.

PFKeys assignment:

12-position table: each position corresponds 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.

## NOTE

If there is no input, the database characteristics don't change.

Any field left blank defaults to the current option selection.

The number of gaps cannot be specified by this procedure.

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

It is thus highly recommended that you set this parameter to '0' or leave it blank (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.

## **RESY - Description of Steps**

### NOTE

The default value of the &ARCHJRNL parameter is NULLFILE. For a standard operation of the database, this parameter must contain the DSNNAME of the archived transactions journal, in the following format:

```
ARCHJRNL='$INDUN..$BASE.PJ(0)'
```

to be able to retrieve backed-up transactions ('REC' is required on the parameter procedure line).

### USER INPUT RECOGNITION: PTU004

Code	Physical name	Type	Label
CARTE	SYSIN	Input	Parameter
PAC7MB	&&RESYMB	Output	Parameter
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data

Code	Physical name	Type	Label
PAC7DD		Input	Batch procedures authorization option

Return code:

- 8: no batch procedure authorization

#### VALIDATION OF JOURNAL EXISTENCE: IDCAMS

LISTCAT of the journal file (AJ).

Return codes:

.0 : The Journal file exists

.OTHER: The Journal file does not exist

#### VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed only if the Journal file exists.

Code	Physical name	Type	Label
PAC7MB	&&RESTMB	Input	User transactions
BVP7AE	&INDSV..BVPAE	Input	Error messages file
B7AJ\$BASE	&INDUV..&BASE.AJ	Input	Journal file
PAC7EU		Report	(only if the journal was not saved)

Return codes:

0: The Journal file was archived.

8: The Journal file was not archived (no REST step is executed).

#### DATABASE POSITIONING: PTU402

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

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV..&BASE.AR	Output	Data file
BVP7AE	&INDSV..BVPAE	Input	Error messages



Code	Physical name	Type	Label
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7MB	&&RESYMB	Input	User transactions
PAC7PS	&&PAC7PS	Output	Work file (2 records, length=144)
PAC7GZ		Report	Backup report

#### DATABASE AVAILABILITY - TRANSACTION RETRIEVAL: PTU420

This step is executed if the Journal file has been archived. It updates the first record of the Data file.

This step is REQUIRED to obtain a consistent database.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7MB	&&RESTMB	Input	User transactions
B7AR\$BASE	&INDUV..&BASE.AR	Input/Output	Development Database data
PAC7JO	&ARCHJRNL	Input	Journal to apply
PAC7PS	&&PAC7PS	Input	Work file
PAC7OJ	&&PAC7OJ	Output	Update transactions (length=170)
PAC7EU		Report	Retrieval report

Return codes:

- 0: Transactions to retrieve
- 4: No transactions to retrieve or error on user input.

In case of abend, the update cannot be performed.

#### UPDATE OF THE DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV..&BASE.AR	Output	Development Database data
B7AN\$BASE	&INDUV..&BASE.AN	Output	Development Database index
B7AY\$BASE	&INDUV..&BASE.AY	Output	Development Database extension

Code	Physical name	Type	Label
B7AJ\$BASE	DUMMY	Output	Development Database Journal
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GY	&INDSV..BVP GY	Input	Administration Database extension
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
B7DC\$BASE	&DSMS..&BASE.DC	Input	DSMS file of Development Database Elements
B7D3\$BASE	&DSMS..&BASE.D3	Input	DSMS index of Development Database Elements
PAC7ME	DUMMY	Input	Work file
PAC7MV	&&PAC7OJ	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

- 0: OK, no error
- 2: Warning
- 4: Critical error

### RESY - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      - "SYSTEM" RELOADING RESTORATION COMPLEMENT -
/**
/** -----
/**
/**
/** INPUT

```

```

/** COL 2      : "Y"
/** COL 10-11  : INITIAL LANGUAGE CODE (FR, EN, ..)
/** COL 12     : "1" INHIBITION OF TRANSACTION LOG
/** COL 14-16  : "REC" FOR RECOVERY OF ARCHIVED TRANSACTIONS
/** COL 17-20  : 4 CHARACTERS TO BE DISPLAYED ON ALL
/**           : SCREEN OF THE PRODUCT
/** COL 21-24  : "NNNN" MAXIMUM NUMBER OF SEARCH ACCESSES
/**           : TO THE DATABASE(LISTS)-(DEFAULT VALUE:300)
/** COL 25     : "U" (DEFAULT VALUE) : IMPLICIT UPDATE
/**           : "N" EXPLICIT UPDATE
/** COL 26-29  : CKECKPOINT FREQUENCY
/** COL 36-47  : PF-KEYS SIGNIFICATIONS
/** COL 79     : BACKUP FILES DISPATCH
/**           : "N" (DEFAULT VALUE) : NO DISPATCH (1 FILE)
/**           : "D" : DISPATCH (2 FILES)
/**
/** IF THE JOURNAL FILE OF TRANSACTIONS ON DISK (AJ) IS NOT
/** REINITIALIZED, THE RESTORE CHAIN IS NOT EXECUTED.
/** IT IS THEREFORE NECESSARY TO EXECUTE THE ARCH
/** PROCEDURE FIRST.
/**
/** -----
/**
/**BVPRESY  PROC BASE=$BASE,          CODE OF VAPAC DATABASE
/**          INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
/**          INDSN='$INDSN',          INDEX OF SYSTEM NO VSAM FILES
/**          INDUN='$INDUN',          INDEX OF USER NO VSAM FILES
/**          INDUV='$INDUV',          INDEX OF USER VSAM FILES
/**          DSMS='$DSMS',            INDEX OF DSMS VSAM FILES
/**          SPAAJ='(TRK,(10,5),RLSE)', JOURNAL SPACE
/**          UWK=$UWK,                WORK UNIT
/**          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
/***:       SYSTCAT='$SCAT',          VSAM SYSTEM CATALOG
/***:       VSAMCAT='$VCAT',         VSAM USER CATALOG
/**          OUT='$OUT',              OUTPUT CLASS
/**          ARCHJRNL=NULLFILE,       DSN OF JOURNAL TO APPLY
/**          RESLIB='$RESLIB',        IMS RESLIB
/**          PROCLIB='$PRCLIB',       IMS PROCLIB
/**          DBDLIB='$DBDLIB',        LIBRARY OF DBD'S
/**          PSBLIB='$PSBLIB',        LIBRARY OF PSB'S
/**          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
/**          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
/*******
/**INPUT    EXEC PGM=DFSRR00,REGION=$REGSIZ,
/**          PARM=(DLI,BVPTU004,PTU004$SUG,&BUF,
/**          &SPIE&TEST&EXCPVR&RST,&PRLD,
/**          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
/**          &IRLM)
/**STEPLIB  DD DSN=&RESLIB,DISP=SHR
/**          DD DSN=&STEPLIB,DISP=SHR
/**          DD DSN=$BCOB,DISP=SHR
/**DFSRESLB DD DSN=&RESLIB,DISP=SHR
/**IMS      DD DSN=&PSBLIB,DISP=SHR
/**          DD DSN=&DBDLIB,DISP=SHR
/***:STEPCAT DD DSN=&SYSTCAT,DISP=SHR

```

```

//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//CARTE DD DDNAME=SYSIN
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&RESYMB,DISP=(,PASS),
//          UNIT=UWK,SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//*-----
//EXISAJ EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(LI&BASE.AJ),DISP=SHR
//*-----
//PTU380 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU380,PTU380$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=((0,NE,INPUT),(00,NE,EXISAJ))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB DD DSN=&&RESYMB,DISP=(OLD,PASS)
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR

```

```

//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7EU DD SYSOUT=&OUT
//*-----
//PTU402 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU402,PTU402$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM),
// COND=((0,NE,INPUT),(00,NE,PTU380))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB DD DSN=&&RESYMB,DISP=(OLD,PASS)
//BVP7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP GU,DISP=SHR
//PAC7PS DD DSN=&&PAC7PS,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
// DCB=(RECFM=FB,LRECL=144,BLKSIZE=144)
//PAC7GZ DD SYSOUT=&OUT
//*-----
//PTU420 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU420,PTU420$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM),
// COND=((00,NE,INPUT),(00,NE,PTU380),(0,NE,PTU402))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,

```

```

//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7MB   DD DSN=&&RESYMB,DISP=(OLD,DELETE)
//PAC7JO   DD DSN=&ARCHJRNL,DISP=OLD,DCB=BLKSIZE=6179
//PAC7OJ   DD DSN=&&PAC7OJ,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=170,BLKSIZE=27880),
//          SPACE=&SPAAJ
//PAC7PS   DD DSN=&&PAC7PS,DISP=(OLD,DELETE)
//PAC7EU   DD SYSOUT=&OUT
//*-----
//PACA15   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPACA15,PACA15$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=((0,NE,INPUT),(0,NE,PTU380),(0,NE,PTU402),(0,NE,PTU420))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE DD DUMMY
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7D3$BASE DD DSN=&DSMS..&BASE.D3,DISP=SHR
//B7DC$BASE DD DSN=&DSMS..&BASE.DC,DISP=SHR
//PAC7IE   DD SYSOUT=&OUT

```

```
//PAC7IF DD SYSOUT=&OUT
//PAC7ME DD DUMMY,DCB=BLKSIZE=353
//PAC7MV DD DSN=&&PAC70J,DISP=(OLD,DELETE)
//PAC7RB DD DUMMY,DCB=BLKSIZE=80
//PAC7RY DD DUMMY,DCB=BLKSIZE=310
//
```

---

## ARCH - Journal Archival

### ARCH - Introduction

This procedure saves the Journal file as a sequential file, and re-initializes it both logically and physically.

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

The archived transactions 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 CONDITIONS

On-line access must be closed.

### ABNORMAL EXECUTIONS

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

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

### NOTE

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

### ARCH - Input / Processing / Results

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

This procedure includes specific optional input to:

- Deactivate previously archived transactions that are considered obsolete.
- Signal the absence of previously archived transactions in input.
- Signal the unavailability of the Data file (AR) in input.
- Request only the re-initialization of the transaction file.

The structure of this input is as follows:

Position	Length	Value	Meaning
2	1	'S'	Line code
3	4	nnnn	Session number
7	8	ccyymmdd	OR date up to which the user requests deactivation
15	1	'I'	Absence of previously archived transactions
16	1	'D'	Data file unavailable
17	1	'J'	Re-initialization without backup, the transactions already archived are NOT retrieved in output.

The session number and the date are exclusive. They are ignored if the absence of input transactions is signalled (refer to Section Recommendations).

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

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

### WARNING

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

If an error occurs on one of the options, a 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, some information in the Specifications Dictionary is sometimes lost, thus preventing the execution of the archiving and restoration 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 backup procedure will not take the Data file (AR) into account. However, the restoration procedure must be executed afterward, since under these conditions, the backup 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 backup procedure formats an empty Journal file. The restoration procedure may then be executed (not compulsory). In this case, the content of the journal file (AJ) is lost.
- If the transactions Back-up file is lost or destroyed, an 'T' must be entered in column 15. As a result, the archiving procedure reformats a new sequential backup file of (archived) transactions.

If one of these columns is accidentally set, and if the archiving procedure is executed while 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 (-1) and (0) to obtain (+1).
- 'D' in col. 16: The archiving 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, (+1 version 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 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 session number.

### **ARCH - Description of Steps**

INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

ARCHIVAL OF JOURNAL FILE: PTU300

This step:

- writes obsolete transactions to be purged on 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.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7JP	&INDUN..&BASE.PJ(0)	Input	Previously archived transactions
B7AJ\$BASE	&INDUV..&BASE.AJ	Input	Development Database journal to re-initialize
PAC7MB	&&ARCHMB	Input	User transactions
PAC7BM	&&ARCHBM	Output	User transactions
B7AR\$BASE	&INDUV..&BASE.AR	Input/Output	Development Database data
PAC7PJ	&INDUN..&BASE.PJ(+1)	Output	Archived update transactions
PAC7PQ	DUMMY	Output	Deactivated transactions (length=170): modify the file name in order to keep the transactions
PAC7EU		Report	Output report
PAC7DD		Report	Batch procedures authorization option

Return codes:

- 0: No error detected on the files,
- 4: Error in journal record (Date or session number not numeric)
- 8: No access authorization for batch procedure, OR: invalid database; in this case, restart the procedure with 'D' in column 16 of the user input .
- 12: Input-output error on a file.

#### CREATION OF THE JOURNAL FILE: IDCAMS

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

Code	Physical name	Type	Label
	&INDUV.&BASE.AJ	Output	Development Database journal

#### RE-INITIALIZATION OF THE JOURNAL FILE: PTU320

This step executes the following:

- Creates the first record in the Journal file,
- Re-initializes the Data file flag with the Journal file's address.

Code	Physical name	Type	Label
PAC7BM	&&ARCHBM	Input	User transactions
B7AR\$BASE	&INDUV.&BASE.AR	Input/Output	Data file
BVP7AE	&INDSV..BVPAE	Input	Error messages file
B7AJ\$BASE	&INDUV.&BASE.AJ	Output	Journal file to re-initialize
PAC7EU		Report	Re-initialization report

Return codes:

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

If the archival and backup procedures are grouped into one job, the PTU320 return code can be tested in order to condition the execution of the backup programs.

#### DELETION OF ARCHIVAL IF DATABASE INCONSISTENCY: IEFBR14

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

Code	Physical name	Type	Label
DDPJ	&INDUN..&BASE.PJ(+1)	Input	Archived update transactions

## ARCH - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*          - ARCHIVAL OF THE JOURNAL -
/*
/* -----
/*
/* INPUT      : COMMAND FOR DEACTIVATION OF ARCHIVED
/*             TRANSACTION
/* COL 2      : "S"
/* COL 3 TO 6 : SESSION NUMBER
/* COL 7 TO 14 : DATE (CCYYMMDD)
/* COL 15     : " " PRESENCE OF ARCHIVED TRANSACTION FILE
/*             : "I" ABSENCE OF ARCHIVED TRANSACTION FILE
/* COL 16     : " " PRESENCE OF DATA FILE (AR)
/*             : "D" ABSENCE OF DATA FILE (AR)
/* COL 17     : " " ARCHIVAL AND REINITIALIZATION
/*             : "J" REINITIALIZATION WITHOUT ARCHIVAL
/*
/* IN THE ABSENCE OF INPUT (OR ERROR ON A COMMAND PARAM.)
/* NO DEACTIVATION WILL TAKE PLACE, HOWEVER ARCHIVAL AND
/* REINITIALIZATION WILL BE EXECUTED NORMALLY.
/*
/* TRANSACTIONS WHOSE SESSION (DATE) IS PRIOR OR EQUAL TO
/* THE SESSION (DATE) INDICATED ARE NOT KEPT. THEY ARE
/* RECOVERED IN THE FILE OF DEACTIVATED TRANSACTION.
/*
/* -----
//BVPARCH    PROC BASE=$BASE,          CODE OF VAPAC DATABASE
//*:         SYSCAT='$VCAT',          VSAM SYSTEM CATALOG
//*:         VSAMCAT='$SCAT',        VSAM USER CATALOG
//          OUT='$OUT',              OUTPUT CLASS
//          VOLS='$VOLUN',           VOLUME OF ARCHIVED TRANSACTIONS
//          INDSN='$INDSN',          INDEX OF SYSTEM NO VSAM FILES
//          INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
//          INDUN='$INDUN',          INDEX OF USER FILES
//          INDUV='$INDUV',          INDEX OF USER VSAM FILES
//          SPAPJ='(TRK,(020,10),RLSE)', SPACE FOR TRANSACTION FILES
//          SPAAJ='(CYL,(3,2),RLSE)', SPACE 'AJ' (OSAM DATABASE)
//          UNITS='$UNITUN',         TRANSACTION FILE UNIT
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          DSCB='$DSCB',            DSCB MODEL FILE
//          PSBLIB='$PSBLIB',        LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',        LIBRARY OF DBD'S
//          RESLIB='$RESLIB',        IMS RESLIB
//          PROCLIB='$PRCLIB',       IMS PROCLIB
//          UWK=$UWK,                WORK UNIT
//          BUF=40, SPIE=0, TEST=0, EXCPVR=0, RST=0, PRLD=, SRCH=0,

```

```

//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&ARCHMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//*-----
//PTU300 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU300,PTU300$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP GU,DISP=SHR
//PAC7JP DD DSN=&INDUN..&BASE.PJ(0),DISP=OLD
//PAC7PJ DD DSN=&INDUN..&BASE.PJ(+1),DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=SER=&VOLS,
//          SPACE=&SPAPJ,
//          DCB=(&DSCB,RECFM=FB,LRECL=170,BLKSIZE=27880)

```

```

//PAC7MB DD DSN=&&ARCHMB,DISP=(OLD,PASS)
//PAC7BM DD DSN=&&ARCHBM,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7PQ DD DUMMY,DCB=BLKSIZE=170
//PAC7DD DD SYSOUT=&OUT
//PAC7EU DD SYSOUT=&OUT
//*-----
//OSAM1 EXEC PGM=IDCAMS,COND=(04,LT,PTU300)
//SYSIN DD DSN=&INDUN..$BASE.SY(DL&BASE.AJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAM2 EXEC PGM=IEFBR14,COND=(04,LT,PTU300)
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,UNIT=&UNITS,
// DISP=(,CATLG,DELETE),VOL=SER=&VOLS,
// DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
// SPACE=&SPAAJ
//*-----
//PTU320 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU320,PTU320$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=(04,LT,PTU300)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7BM DD DSN=&&ARCHBM,DISP=(OLD,DELETE)
//PAC7EU DD SYSOUT=&OUT
//*-----
//DEL PJ EXEC PGM=IEFBR14,COND=(8,NE,PTU300)
//DDPJ DD DSN=&INDUN..&BASE.PJ(+1),DISP=(OLD,DELETE)
//

```

---

## REOR - Reorganization

### REOR - Introduction

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

It uses one or two backup files of the development Database (if 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 operating principle of this procedure is to rebuild the different indexes associated with all data using the 'image' of each Data Element. It makes the best of the system performance features since it separates historical (frozen) sessions from the current session and sorts the data in the order of the most frequent access. This makes it possible to achieve a significant reduction of the number of indexes and data items.

The REOR procedure may be used in two cases:

- When part of the data was deleted because of a malfunction or system failure, and no other procedure can be used (in particular, deletion of the AN Index file),
- When the database is to be purged of the following:
  - Obsolete libraries and/or sessions;
  - Entities not used in the database;

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

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

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

- Deletion of unused Production sessions (PEI Function)
- Deletion of entities not associated to a specific use, determined by the unused-entity extraction utility, EXPU (see the PACX procedure in 'the Developer's Procedures' manual).

### EXECUTION CONDITIONS

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.

### ABNORMAL EXECUTIONS

Refer to chapter 'Overview', subchapter 'Abnormal Ending'.

As specified in the '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.

### **REOR - Input / Processing / Results**

A '\*' line with user code and password.

Specific user input for the procedure (optional) which enables to specify:

- Libraries to be purged,
- sessions to be purged or to be kept,
- users to be purged,
- entities to be purged (notion of cross-references taken into account),
- a report of the duplicate indexes of the REOR procedure.

Position	Length	Value	Meaning
2	1	'B'	Library purge
3		bbb	Library code(s)
			up to 23 library codes per line

Maximum number of libraries to be purged: 300.

Position	Length	Value	Meaning
2	1	'V'	Purge frozen sessions
		'S'	Save frozen sessions
			'V' and 'S' lines are not compatible
3		ssss	Session number(s): up to 17 session numbers per line, entered without any separator.



Maximum number of sessions indicated on the request: 999.

Maximum number of frozen sessions indicated in a database: 7,500.

Position	Length	Value	Meaning
2	1	'U'	Purge users (up to 9 user codes per line)

Position	Length	Value	Meaning
2	1	'E'	Physical purge of entities (transactions provided by EXPU)
3			Entity (required)
	1		.Type
	2		.UE call type (if Type "\$")
6	30		Code of the entity to be purged (this code can be generic)
36	3		Library code (required)
			5 groups of type/code entity/Lib. possible per 'E'-type line
39		' ' or 'N'	Purge of the Library and of its sub-Libraries
		'Y'	Purge of the Library alone

One line per entity. The 'List of 'purged' entities' signals what has been done.

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

Position	Length	Value	Meaning
2	1	'P'	Physical purge of unformatted entities
3			Entity (required)
	1		Type
	2		UE call type (if type '\$')
6	30		Code of the entity to be purged (required, complete identifier)
36	2		Entity Description type (required, value '00' not authorized)
		'nn'	Purge of the 'nn' descriptions of the entity

Position	Length	Value	Meaning
		'**'	Purge of all Descriptions of the entity
38	3		Library code (required)
		'***'	All Libraries of the entity
41	1		Flag for purge of lower-level Libraries
		' ' or 'N'	Purge of the Library and its sub-Libraries
		'Y'	Purge of the Library alone
42	4		Session lower bound (one of the two bounds is required)
		' '	Purge from the lower bound
		'nnnn'	Purge from session 'nnnn' (included)
		'****'	Purge of all sessions
46	4		Session higher bound (one of the two bounds is required)
		' '	Purge until the higher bound
		'mmmm'	Purge until session 'mmmm' (included)
		'****'	Purge in all sessions

One line per 'unformatted'-type entity.

Note on 'E- and P-type' purge requests: The execution of the REOR procedure processes a maximum number of 2,500 instances of an entity type other than a User Entity and a number of 1,000 instances of the User Entity type.

Position	Length	Value	Meaning
2	1	'D'	List of duplicate indexes of the REOR procedure
3	1	' '	no report of duplicate indexes
		'1'	Report of duplicate indexes
		'2'	Report of PMS calls which might be incoherent
		'3'	Reports of values '1' and '2'

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 REPORT

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

### 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 presents a number of characteristics which the user should be aware of:

- The step that rebuilds the Index file uses a large amount of CPU time.
- 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.

## **REOR - Description of Steps**

INPUT RECOGNITION: PTU001

VALIDATION OF USER INPUT: PTU2CL

This step validates user input and sets a return code when an error is detected.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the development Database
PAC7MB	&&REOR MB	Input	Input work file
PAC7BM	&&REOR BM	Output	Formatted records
PAC7PU	&&PU	Output	Entity records to be purged (length=44)
PAC7EE		Report	Control report
PAC7DD		Report	Batch procedures authorization option
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return code(s):

- 0: OK
- 4: Error on user input
- 8: No authorization for Batch procedure.

#### CREATION OF IDCAMS EXTENSION DATA FILE

This step executes a DELETE/DEFINE of the working file which contains the extension data WKREOR

Code	Physical name	Type	Label
	&INDUV..WKREOR	Output	Working file

#### CREATION OF IDCAMS EXTENSION DATA FILE

This step executes a DELETE/DEFINE of the working file which contains the complementary extension data WYREOR

Code	Physical name	Type	Label
	&INDUV..WYREOR	Output	Working file

### 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 input file, which contains the data, i.e. PC(0)). It then formats the key of each record selected for the subsequent sort.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7BM	&&REORBM	Input	User transactions
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the development Database
PAC7PY	&INDUN..&BASE.PY(0)	Input	Sequential image of the extension data of the development Database
PAC7PR	&&QS	Output	Formatted records (length=176 size= ND)
PAC7NX	&INDUV..WKREOR	Output	Long data
PAC7NY	&INDUV..WYREOR	Output	Unformatted data
PAC7AU	&&AU	Output	PR image (length=153)
PAC7EE		Report	Report on statistics retrieval

### SORT OF DATA: SORT

Sorting criteria: SRTREO1 member of the SY PDS.

Code	Physical name	Type	Label
SORTIN	&&QS	Input	Formatted records
SORTOUT	&&SQ	Output	Sorted records (length=176)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### PURGE: PTU210

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

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7PR	&&SQ	Input	Sorted records
PAC7BM	&&REORBM	Input	User transactions
PAC7PU	&&PU	Input	Entity records to be purged
PAC7QS	&&QS	Output	Purged records (length=176, size=ND)
PAC7UM	&&UM	Output	Macro-structure call lines (length=176)
PAC7EE		Report	Library and session purge report
PAC7EK		Report	Entity-purge report
PAC7EB		Report	Technical report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0: OK
- 8: Overload of capacity

The following steps are executed only if the return code is 0.

#### INDEX REBUILDING: PTU220

This step executes two types of procedures:

- Reconstruction of the indexes using the data.
- Separation of the current session and the frozen sessions.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages file
PAC7BM	&&REORBM	Input	User transactions
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7UR	&&QS	Input	Purged data

Code	Physical name	Type	Label
PAC7NX	&INDUV..WKREOR	Input	Long data
PAC7UM	&&UM	Input	Macro-structure call lines
PAC7PA	&&PA	Output	Data from frozen sessions (length=153 size=NH)
PAC7PB	&&PB	Output	Data from current session (length=153 size=NC)
PAC7PC	&&PC	Output	First data records (length=153)
PAC7AN	&&AN	Output	Intermediate index file (length=60 size=NI)
PAC7MR		Input/Output	Macro-structure call lines
PAC7EE		Report	Index-building report

### SORT OF INDEXES: SORT

Sort criterion: SRTREO2 member of SY PDS.

Code	Physical name	Type	Label
SORTIN	&&AN	Input	Temporary index
SORTOUT	&&NA	Output	Sorted indexes
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### EXTENSION DATA PROCESSING: PTU226

Code	Physical name	Type	Label
PAC7NY	&INDUV..WYREOR	Input	Unformatted data
PAC7PA	&&PA	Input	Data from frozen sessions
PAC7PB	&&PB	Input	Data from current session
PAC7PC	&&PC	Input	First data records
PAC7QA	&&QA	Output	Data from frozen sessions (length=153)
PAC7QB	&&QB	Output	Data from current session (length=153)
PAC7QC	&&QC	Output	First data records (length=153)

Code	Physical name	Type	Label
PAC7QY	&&QY	Output	Long data (length=1,018)

### FORMATTING SEQUENTIAL IMAGE :

IEBGENER

MERGE: PTU240

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

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7AN	&&NA	Input	Sorted index
PAC7AU	&&AU	Input	PR image
PAC7BM	&&REORBM	Input	User transactions
PAC7PA	&&QA	Input	Data from frozen sessions
PAC7PB	&&QB	Input	Data from current session
PAC7PC	&&QC	Input	First data records
PAC7QY	&&QY	Input	Extension data
PAC7CP	&INDUN..&BASE.PC(+1)	Output	Sequential image of the development Database
PAC7PD	&INDUN..&BASE.PD(+1)	Output	Sequential image of the development Database
PAC7PY	&INDUN..&BASE.PY(+1)	Output	Sequential image of the development Database
PAC7IE		Report	Building of Logical database

### **REOR - Execution JCL**

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**              - REORGANIZATION OF THE DATABASE -
/**
/** -----
/**
/** THE REOR PROCEDURE MAY BE USED IN TWO CASES:
/** . WHEN PART OF THE DATA WAS DELETED BECAUSE OF A MAL-
/**   FUNCTION 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;
/**
/** -----
/**
/**BVPREOR PROC BASE=$BASE, CODE OF VAPAC DATABASE
/** INDSV='$INDSV', INDEX OF VSAM SYSTEM FILES
/** INDSN='$INDSN', INDEX OF NO VSAM SYSTEM FILES
/** INDUN='$INDUN', INDEX OF NO VSAM USER FILES
/** INDUV='$INDUV', INDEX OF VSAM USER FILES
/** OUT='$OUT', OUTPUT CLASS
/** CYL=(10,1)', SORT WORKS SIZE
/** SPAPC=(TRK,(300,10))', VA PAC DATABASE BACKUP
/** SPAPD=(TRK,(300,10))', VA PAC DATABASE BACKUP 2
/** SPAPY=(TRK,(300,10))', VA PAC DATABASE BACKUP 3
/** STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
/** SORTLIB='$BIBT', SORT LIBRARY
/** DSCB='$DSCB', DSCB MODEL FILE
/** UWK=$UWK, WORK FILE UNIT
/** VOLS='SER=$VOLUN', BACKUP FILE VOLUME
/** UNITS=$UNITUN, BACKUP FILE UNIT
/** VOLS2='SER=$VOLUN', 2ND BACKUP FILE VOLUME
/** VOLS3='SER=$VOLUN', 3RD BACKUP FILE VOLUME
/** UNITS2=$UNITUN, 2ND BACKUP FILE UNIT
/** UNITS3=$UNITUN, 3RD BACKUP FILE UNIT
/***: SYSTCAT='$SCAT', VSAM SYSTEM CATALOG
/** PSBLIB='$PSBLIB', LIBRARY OF PSB'S
/** DBDLIB='$DBDLIB', LIBRARY OF DBD'S
/** RESLIB='$RESLIB', IMS RESLIB
/** PROCLIB='$PRCLIB', IMS PROCLIB
/** BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
/** CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
/**-----
/**INPUT EXEC PGM=BVPTU001
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/** DD DSN=$BCOB,DISP=SHR
/**CARTE DD DDNAME=SYSIN
/**PAC7MB DD DSN=&&REORMB,DISP=(,PASS),
/** UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
/** DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
/**-----
/**PTU2CL EXEC PGM=DFSRR00,REGION=$REGSIZ,
/** PARM=(DLI,BVPTU2CL,PTU2CL$SUG,&BUF,
/** &SPIE&TEST&EXCPVR&RST,&PRLD,
/** &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
/** &IRLM)
/**STEPLIB DD DSN=&RESLIB,DISP=SHR
/** DD DSN=&STEPLIB,DISP=SHR
/** DD DSN=$BCOB,DISP=SHR
/**DFSRESLB DD DSN=&RESLIB,DISP=SHR
/**IMS DD DSN=&PSBLIB,DISP=SHR
/** DD DSN=&DBDLIB,DISP=SHR
/***:STEPCL DD DSN=&SYSTCAT,DISP=SHR

```

```

//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,2,,CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7PC DD DSN=&INDUN..&BASE.PC(0),DISP=SHR
//PAC7EE DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&REORMB,DISP=(OLD,PASS)
//PAC7BM DD DSN=&&REORBM,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7PU DD DSN=&&PU,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=50,BLKSIZE=4100)
//*-----
//DEFINE EXEC PGM=IDCAMS,COND=(0,NE,PTU2CL)
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(DFWKREOR),DISP=SHR
//*-----
//DEFINE EXEC PGM=IDCAMS,COND=(0,NE,PTU2CL)
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(DFWYREOR),DISP=SHR
//*-----
//PTU200 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU200,PTU200$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=(00,NE,PTU2CL)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT

```

```

//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7BM DD DSN=&&REORBM,DISP=(OLD,PASS)
//PAC7EE DD SYSOUT=&OUT
//PAC7NX DD DSN=&INDUV..WKREOR,DISP=SHR
//PAC7NY DD DSN=&INDUV..WYREOR,DISP=SHR
//PAC7PC DD DSN=&INDUN..&BASE.PC(0),DISP=SHR
//PAC7PY DD DSN=&INDUN..&BASE.PY(0),DISP=SHR
//PAC7AU DD DSN=&&AU,DISP=(,PASS),
// UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PR DD DSN=&&QS,DISP=(,PASS),
// UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//*-----
//SORTQS EXEC PGM=SORT,COND=((0,NE,PTU2CL),(00,NE,PTU200))
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN DD DSN=&&QS,DISP=(OLD,DELETE)
//SORTOUT DD DSN=&&SQ,DISP=(,PASS),
// UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//SYSIN DD DSN=&INDSN..BVPSY(SRTRE01),DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//*-----
//PTU210 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU210,PTU210$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM),
// COND=((00,NE,PTU2CL),(00,NE,PTU200),(0,NE,SORTQS))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR

```

```

//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7BM DD DSN=&&REORBM,DISP=(OLD,PASS)
//PAC7PR DD DSN=&&SQ,DISP=(OLD,DELETE)
//PAC7PU DD DSN=&&PU,DISP=(OLD,DELETE)
//PAC7QS DD DSN=&&QS,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//PAC7UM DD DSN=&&UM,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=176,BLKSIZE=27984)
//PAC7EB DD SYSOUT=&OUT
//PAC7EE DD SYSOUT=&OUT
//PAC7EK DD SYSOUT=&OUT
//*-----
//PTU220 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU220,PTU220$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=((00,NE,PTU2CL),(00,NE,PTU200),(0,NE,SORTQS),
//          (00,NE,PTU210))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7BM DD DSN=&&REORBM,DISP=(OLD,PASS)
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR

```

```

//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7EE DD SYSOUT=&OUT
//PAC7UM DD DSN=&&UM,DISP=(OLD,PASS)
//PAC7UR DD DSN=&&QS,DISP=(OLD,PASS)
//PAC7AN DD DSN=&&AN,DISP=(,PASS),
// UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=60,BLKSIZE=27960)
//PAC7MR DD UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=144,BLKSIZE=6336)
//PAC7PA DD DSN=&&PA,DISP=(,PASS),
// UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PB DD DSN=&&PB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PC DD DSN=&&PC,DISP=(,PASS),UNIT=&UWK,
// SPACE=(TRK,1,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=153)
//PAC7NX DD DSN=&INDUV..WKREOR,DISP=SHR
//*-----
//SORTAN EXEC PGM=SORT,
// COND=((00,NE,PTU2CL),(0,NE,PTU200),(00,NE, SORTQS),
// (00,NE,PTU210),(0,NE,PTU220))
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN DD DSN=&&AN,DISP=(OLD,PASS)
//SORTOUT DD DSN=&&NA,DISP=(,PASS),
// UNIT=&UWK,SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=60,BLKSIZE=27960)
//SYSIN DD DSN=&INDSN..BVPSY(SRTREO2),DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//*-----
//PTU226 EXEC PGM=BVPTU226,COND=((0,NE,PTU2CL),(0,NE,PTU200),
// (0,NE, SORTQS),(0,NE,PTU210),(0,NE,PTU220),(0,NE, SORTAN))
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7NY DD DSN=&INDUV..WYREOR,DISP=SHR
//PAC7PA DD DSN=&&PA,DISP=(OLD,DELETE)
//PAC7PB DD DSN=&&PB,DISP=(OLD,DELETE)
//PAC7PC DD DSN=&&PC,DISP=(OLD,DELETE)
//PAC7QA DD DSN=&&QA,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7QB DD DSN=&&QB,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(CYL,&CYL,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7QC DD DSN=&&QC,DISP=(NEW,PASS),UNIT=&UWK,
// SPACE=(TRK,1,RLSE),
// DCB=(RECFM=FB,LRECL=153,BLKSIZE=153)

```

```

//PAC7QY DD DSN=&&QY,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=1018,BLKSIZE=27486)
//IEBGN1 EXEC PGM=IEBGENER,
//          COND=((00,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
//          (0,NE,PTU210),(0,NE,PTU220),(0,NE,SORTAN))
//*-----
//SYSPRINT DD DUMMY
//SYSIN DD DUMMY
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=&INDUN..&BASE.PC(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,
//          VOL=&VOLS,
//          SPACE=&SPAPC,
//          DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//IEBGN2 EXEC PGM=IEBGENER,
//          COND=((00,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
//          (0,NE,PTU210),(0,NE,PTU220),(0,NE,SORTAN))
//*-----
//SYSPRINT DD DUMMY
//SYSIN DD DUMMY
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=&INDUN..&BASE.PD(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS2,
//          VOL=&VOLS2,
//          SPACE=&SPAPD,
//          DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//IEBGN3 EXEC PGM=IEBGENER,
//          COND=((00,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
//          (0,NE,PTU210),(0,NE,PTU220),(0,NE,SORTAN))
//*-----
//SYSPRINT DD DUMMY
//SYSIN DD DUMMY
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=&INDUN..&BASE.PY(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS3,
//          VOL=&VOLS3,
//          SPACE=&SPAPY,
//          DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//*-----
//PTU240 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU240,PTU240$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=((00,NE,PTU2CL),(0,NE,PTU200),(0,NE,SORTQS),
//          (0,NE,PTU210),(0,NE,PTU220),(0,NE,SORTAN))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR

```

```

//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7IE   DD SYSOUT=&OUT
//PAC7BM   DD DSN=&&REORBM,DISP=(OLD,DELETE)
//PAC7AU   DD DSN=&&AU,DISP=(OLD,DELETE)
//PAC7AN   DD DSN=&&NA,DISP=(OLD,DELETE)
//PAC7PA   DD DSN=&&QA,DISP=(OLD,DELETE)
//PAC7PB   DD DSN=&&QB,DISP=(OLD,DELETE)
//PAC7PC   DD DSN=&&QC,DISP=(OLD,DELETE)
//PAC7QY   DD DSN=&&QY,DISP=(OLD,DELETE)
//PAC7CP   DD DSN=*.IEBGN1.SYSUT2,DISP=(OLD,,DELETE)
//PAC7PD   DD DSN=*.IEBGN2.SYSUT2,DISP=(OLD,,DELETE)
//PAC7PY   DD DSN=*.IEBGN3.SYSUT2,DISP=(OLD,,DELETE)
//*-----
//DELETE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DLWCREOR),DISP=SHR
//DELETE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&INDSN..BVPSY(DLWYREOR),DISP=SHR
//

```





---

## Chapter 4. Manager's Utilities

---

### PACX - Extractions

#### PACX - Introduction

The extraction procedure allows to perform various types of data extractions from the VA Pac Database via a PAF extractor (selection criteria).

See chapter 'UPDP - Update from PAF Tables' in 'The Developer's Procedures' manual.

This data is extracted in the form of transactions that can be used in input of the following procedures:

- UPDT
- UPDP
- CPSN (If the optional LCU Partitioned Database Manager utility is available.)

#### EXECUTION CONDITIONS

None since the Database is not directly updated by this procedure.

#### PACX - Input Common to Extractors

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Extraction library code, or target Library code if RMEN with upload
22	4	nnnn	Session number (blank=current ses.)
26	1	T	Session status if Test session
29	4	cccc	Extractor code (1)
33	1	'1'	Formatting for UPDT
		'2'	CPSN : formatting for UPDT with explicit transaction codes
		' '	No formatting for UPDT
34	1	'1'	Formatting for UPDP (PAF)

Position	Length	Value	Meaning
		'2'	CPSN : formatting for UPDP with explicit transaction codes
		' '	No formatting for UPDP (PAF)
35	1	'1'	Formatting for CPSN
		' '	No formatting for CPSN
40	3	ppp	DSMS Product Code
43	6	nnnnnn	DSMS Change number (DSMS Function only)
49	1		Lock processing
		' '	Lock extraction: user code = '*'-line user code
		'1'	No lock extraction
		'2'	Lock extraction: user code = original user code
		'N'	For RMEN only : no extraction of locked entities by an other user
50	1	' '	No transfer of password
		'1'	Password transfer
69	3	bbb	Library code for the '*'-line of the output file(s) (For EXTR, EXLI, and EXUE only)
76	5	nnnnT	Session number for the '*'-line of the output file(s) (For EXTR, EXLI, and EXUE only)

(1) Possible values for the extractor code include:

- EXTR: Extraction of entities (extracted transactions are sorted).
- EXTA: Extraction of entities (extracted transactions are sorted, according to the input identification lines order. So if each request is preceded by a '\*' line, extracted transactions will be sorted in the order of the requests). The formatting is forced to UPDT.
- EXUE: Extraction of user entities
- EXLI: Extraction of libraries or library sub-networks (formatting for UPDP, UPDT or CPSN).
- EXPJ: Extraction of Journal (formatting for CPSN is not possible)
- EXPU: Extraction for purge (formatting for CPSN is not possible)

- RMEN: Extraction of entities for upload/replacement/ recoding (formatting for CPSN is not possible). RMEN is subject to a separate purchase agreement.
- CPSN: comparison of sub-networks or entities.

### IMPORTANT

- One extractor type only for each run: If the procedure detects more than one type of extractors, it will take only the first one into account.
- Formatting for CPSN: This procedure is part of the 'LCU Partitioned Database Manager' optional utility. Its use is therefore subject to a separate purchase agreement.
- Maximum number of input '\*' lines : 1 for RMEN and EXPJ, 400 for EXSN, 1000 for EXTR, EXTA, EXUE and EXPU.

### RESULTS

The PACX procedure produces:

- A report containing the list of executed programs and the number of generated transactions.
- A list of requests with possible associated errors.
- One or several execution reports depending on the type of extractor.

### **Extraction of Archived Transactions**

#### **EXPJ - Introduction:**

The EXPJ procedure has a two-fold action:

- It converts the Journal file into update transactions with possible selection from a range of dates, sessions, libraries, etc.
- It prints out a listing of the contents of the archived Journal file, using the same criteria.

Its main purpose is to retrieve transactions associated with one database in order to update another database.

It is executed on the archived Journal file (PJ).

### EXECUTION CONDITIONS

Password transfer option (\*'-line col. 50 = 1):

. Database access authorization level 4 is required.

#### **EXPJ - User Input:**

User entry specific to this procedure and specifying the extraction characteristics.

Position	Length	Value	Meaning
2	1	'J'	Line code
3	1	'S'	Selection on session number
		'D'	Selection on date
4	1	' '	Chronological sort
		'N'	No chronological sort
5	1	' '	Sort by user
		'N'	No sort by user
6	1	' '	Sort by Library
		'N'	No sort by library
7	1	' '	Sort by session
		'N'	No sort by session
8	8	uuuuuuuu	User code for batch update
16	8	pppppppp	User password
24	4	dddd	Session number: beginning (if 'S')
28	4	ffff	Session number: end (if 'S')
32	8	CCYYMMDD	Date of beginning of select.(if 'D')
40	8	CCYYMMDD	Date of end of selection (if 'D')
48	1		Version of selected transactions
		' '	Selection of all sessions
		'Z'	Selection of current session
		'T'	Selection of frozen session
49	3	'bbb'	Code of selected library
52	5	'ssssT'	Selection of T-type session (test version of frozen session:'ssssT')
57	3	ppp	DSMS Product Code
60	6	nnnnnn	DSMS Change number (Selection by change number-DSMS)
66	6	HHMMSS	Starting time
72	6	HHMMSS	Ending time
80	1	'**'	If selection on user, indicates continuation line

Second user entry if selection on user code :

Position	Length	Value	Meaning
2	1	'J'	Line code
3	1	'*'	Indicates a following line
4	8	uuuuuuuu	User code

### REPORTS

- The list of selection options used,
- The list of selected transactions, if requested.

### RESULTS

In the case of a request for conversion of the Journal entries into transactions, the result of the EXPJ procedure is a sequential file containing all selected transactions.

### **Extraction of Libraries**

#### **EXLI - Introduction:**

The EXLI procedure extracts a complete library or a sub-network of Libraries in a transactions file format.

The file obtained, according to its formatting, can be used as input to the UPDT or UPDP update procedures or to the CPSN Sub-Network Comparison procedure.

### EXECUTION CONDITIONS

If DESIGN entities have been downloaded and have then been locked, they must be uploaded before the extraction to ensure data consistency.

#### **EXLI - User Input:**

No specific line, but as many '\*'-lines as there are libraries to be extracted in the sub-network.

### PRINTED REPORT

The extractor prints:

- A list of extracted libraries with the number of records for each library,
- The details of records extracted for each library.

## Extraction for Purge

### EXPU - Introduction:

The EXPU utility purges:

- unused entities from a database,
- frozen sessions which have been logically cancelled,
- cancelled libraries,
- lines belonging to the users who no longer exist.

Several types of purges are possible:

- 'Logical' purge of entities which have become obsolete;
- 'Logical' purge of unused entities in a particular context;
- 'Physical' purge of entities which have never been used.

### TERMINOLOGY

- Final entities

These are entities which are not used by other entities.

- Free-type cross-reference

Reference whose existence does not prevent deletion of the Definition screen of the Entity on which it is dependent.

### PRINCIPLES

- Logical purge: the EXPU procedure shows the list of entities which have not been used since an indicated frozen session and in a given context.

For these entities, the procedure generates logical deletion transactions of definition and description lines. These transactions can be used as input to the UPDT procedure.

For free-type entities, no deletion transaction is generated: only a message is printed in the report.

- Physical purge: the EXPU procedure gives the list of entities which have never been cross-referenced since they have been created in a given context. For these entities, physical purge transactions are generated. They can be used as input to the REOR procedure.

Limits of physical purge: if a data structure has already been purged, there is no physical purge possible for its segments.

- Purge of frozen sessions: the extraction for purge procedure indicates to the user the sessions which have been logically cancelled.

For these sessions, physical-purge transactions are generated. They can be used as input to the reorganization procedure.

- Purge of libraries: the extraction for purge procedure indicates to the user the libraries which have been cancelled.

For these libraries, physical-purge transactions are generated. They can be used as input to the reorganization procedure.

- Purge of GP lines: the extraction for purge procedure specifies to the user, the users who no longer exist in the administration database, despite the GP lines which still exist in the database.

For these users, physical-purge transactions are generated. They can be used as input to the reorganization procedure.

### EXECUTION CONDITIONS

None.

### **EXPU - User Input:**

One line with the extraction characteristics:

Position	Length	Value	Meaning
2	2	'P '	Line code
3	1		Purge of frozen sessions
		'S'	yes
		' '	no purge
4	1		Purge of libraries
		'B'	yes
		' '	No purge
5	1		Purge of GP lines for users who no longer exist
		'U'	yes
		' '	No purge
6	1		Purge of entities
		'P'	Physical (via the REOR procedure)
		'L'	Logical (via the UPDT/UPDP update)
		' '	No purge
7	1	t	Entity type
8	1	m	Print option for the last entity update: user code and date
9	1	s	Print option for the last session in which the entity is used
10	4	ssss	Session number (L type only) from which the entities should not be used in order to be logically purged

Position	Length	Value	Meaning
14	6	pppppp	Program code where the search stops if the programs are processed Information required if the entity type is 'P' or if it is blank

### COMMENTS

Each entity type may be processed separately. If there is no entity type entered, all the entities are processed except the final entities.

### COMMAND EXAMPLES

\*user\_\_\_passwordLIB

P\_\_\_PE1

Command for physical purge transactions for the data elements in the BIB library sub-network and printing of the last update (user and date).

\*user\_\_\_passwordLIB

P\_\_\_LP112222PROGR

Command for logical deletion transactions for the programs in the BIB library sub-network whose codes are less than or equal to PROGR, and no longer used since the 2222 session with the printing of the last update (user and date) and the last session of use.

\*user\_\_\_passwordLIB

PSBUP\_\_\_\_\_PROGR

Command for the physical purge transactions for all entities in the BIB library sub-network (except the final entities), for logically cancelled frozen sessions, for deleted libraries and GP lines of users who no longer exist.

### PRINTED REPORTS

This procedure prints out:

- the list of entities to be purged logically,
- the list of entities to be purged physically,
- the list of entities copied in the sub-network,
- the list of frozen sessions to be purged physically.



- the list of libraries to be purged physically.
- the list of users whom GP lines are to be purged physically.

## RESULT

The result of this procedure is:

- In the case of a logical purge, a sequential file containing entity deletion transactions to be used as input in the batch updating UPDT or UPDP procedures. These transactions are sorted as follows:
  - by decreasing hierarchical library level,
  - by library,
  - by record type: descriptions, definition screens.
- In the case of a physical purge, purge of frozen sessions, purge of libraries or GP lines, a sequential file of purge transactions to be used as input in the REOR procedure.

Each transaction contains an entity to be purged. For each entity, the following information is included:

- the type of entity,
- the entity code,
- the library code (see section 'Input Recommendations' in subchapter 'Database Reorganization').

## **Standardization Utility**

### **RMEN - Introduction:**

The RMEN procedure is an optional utility. It is subject to a separate purchase agreement.

Through this procedure you can:

- Rename an entity
- Replace an entity with another
- Move an entity to a higher-level library
- Rename and move up an entity simultaneously.

This procedure may be applied to the Dictionary and WorkStation entities.

Its output is a file containing update transactions, which will be used as input to the batch update procedure (UPDT or UPDP).

## EXECUTION CONDITIONS

None, since the Database is not directly updated.

## RMEN - User Input:

Several command lines per entity to be processed:

First line - concerned entity :

Position	Length	Value	Meaning
2	2	'W2'	Line code
4	2		Processing option:
		'MV'	Entity move (UP)
		'RN'	Entity rename
		'MR'	Upward move and rename
		'RP'	Entity replace
6	1	' '	Line type
7	3	ttt	Entity type or local code of a WorkStation entity: D, E, I, O, P, R, S, T, \$nn, Ynn, M, Q, B, V, or DST, DEL ...
10	30	eeee..	Code of entity to be extracted

Second line - environment :

Position	Length	Value	Meaning
2	2	'W2'	Line code
4	2		Processing option - same as line 1
6	1	'E'	Line type
7	3	sss	Source library code (for MOVE)
10	3		For extraction of WorkStation entities: methodology code
		'//A'	SSADM
		'//M'	MERISE
		'//D'	YSM
		'//O'	OMT
		'//F'	IFW
13	3	'ALL'	for 'MV' and 'MR': Selects all occurrences of a UE or all Segments or Reports of a Data Structure (implicit option for 'RN' and 'RP')
16	6	rrrrrr	Parent Data Element code

Third line - new codes:

Position	Length	Value	Meaning
2	2	'W2'	Line code
4	2		Processing option - same as line 1
6	1	'N'	Line type
7	30	nnnn...	New entity code
37	8	gggggggg	For programs and screens, new generated code
45	6	cccccc	For programs, new sequencing code
51	8	eeeeeee	For screens, new map code

Fourth line - selection for REPLACE:

Position	Length	Value	Meaning
2	2	'W2'	Line code
4	2	'RP'	'REPLACE'
6	1	'S'	Line type
7	3		REPLACE: Selection of the types of the entities to be modified
			'DEL': Data Element
			'DBD': Database Block
			'DST': Data Structure
			'SEG': Segment
			'RPT': Report
			'TXT': Text
			'VOL': PDM volume
			'PGM': Program
			'SCR': Screen
			'PIA': P.I.A.
			'MET': Methodology
			'CME': Client Meta-Entity
			'CRL': Client Relationship
			'\$tt': Client User Entity : (tt = type code)
			'\$**': All Client User Entities

Position	Length	Value	Meaning
			'Ytt': Extension User Entity : (tt = type code)
			'Y**': All Extension User Entities
10	30		REPLACE: Codes of entities to be modified (* may be used if you want to specify only the beginning of a code)

Lines for REPLACE (continuation lines for selection):

Position	Length	Value	Meaning
2	2	'W2'	Line code
4	2	'RP'	'REPLACE'
6	1	'*'	Line type
7	3		Selection of types of entities to be modified
10	30		Codes of entities to be modified

Last line (required):

Position	Length	Value	Meaning
2	2	'W2'	Line code
4	2		Processing option - same as line 1
6	1	'/'	Line type

## REQUEST-SEQUENCING REQUIREMENTS

The sequencing of RMEN requests should follow a logical order.

Examples:

- A parent Data Element must be moved to the higher-level library before its child Data Element(s).
- When a Segment is called by another Segment, the called Segment must be moved to the higher-level library before the Segment that is calling it.
- When a macro-structure is called by a batch Program or on-line Screen, it must be moved into the higher-level library before this Program or Screen.

## REQUEST-INPUT REQUIREMENTS

All input is required except:

- The source library code in case of entity renaming (RN) or replacing (RP),
- The new entity code in case of upward move (MV),
- The code of the parent data element (except when a child data element is to be associated with it).

The 'RP' processing type is incompatible with the other processing types.

### EXECUTION CONDITIONS

The source library must belong to the sub-network of the target library.

When an upward move is requested for an entity which already exists in the target library, a warning message appears in the report, but the transaction is still generated.

### PRINTED REPORT

This procedure prints out the following:

- The list of entities processed by RMEN.
- The number of lines extracted for each request.

### RESULT

The output is a sequential file which contains update transactions:

- Creation or modification transactions sorted by:
  - Ascending library hierarchical level,
  - Library,
  - Record type (uses, definition, or description).
- Deletion transactions sorted by:
  - Descending library hierarchical level,
  - Library,
  - Record type (uses, description, definition).

### NOTE

The replacement of entities (RP) does not ensure data consistency.

Example: if you replace a Data Element with another one in a Segment, RMEN does not modify the program lines where this Data Element is used by this Segment, except if you have requested the replacement in programs.

New occurrence codes longer than the initial ones may sometimes cause update transactions to be truncated. However, they will still belong to the flow of update transactions, but will also appear in the validation report with a warning message.

**WARNING:** If not correctly managed, the RMEN procedure may have undesired effects on the Database. Caution is highly recommended when requesting its execution.

### **RMEN - Recommendations and Restrictions:**

Processing in a frozen session is possible. The number of the session is indicated on the '\*' line.

When an error is detected on the '\*' line, the request flow is not processed.

Only one '\*' line is authorized.

### ALL ENTITY TYPES

The MOVE+RENAME (MR) command first moves and then renames. The consequence is that all the entities bearing the same code within the sub-network of libraries equal to or lower than the target library are renamed by the RMEN procedure.

If this result is not satisfactory, it is advised to first run a RMEN/RENAME followed by a UPDT, then a RMEN/MOVE followed by another UPDT execution.

If the entity uses other entities, these used entities must exist in a library whose level is greater than or equal to that of the target library.

When an occurrence is renamed, if it is called on Assigned Text lines,

- it is changed on I-type lines,
- it is not changed on J-type lines.

### DATA STRUCTURES

Renaming a Data Structure causes the renaming of all its Segments.

**WARNING:** An upward move of a Data Structure involves the upward move of all of its Segments contained in the source library in cases where the 'global upward move' field contains 'ALL'. If this field is blank, the Segments remain in the source library.

The existence of the Data Structure in an upper-level library is checked.

## SEGMENTS

These entities can only be moved upward. Their Data Structure must exist in a Library whose level is higher than or equal to that of the target Library.

The existence of a Segment in a library whose level is higher than or equal to that of the target Library is checked, as is that of called Segments, Data Elements, and MERISE Objects and Relationships.

## REPORTS

It is not possible to change a single report code or to replace a single report. It is however possible to perform an upward move of a single report.

However, you can rename, move upward or replace all the reports with the same prefixes (two first characters), you just need to enter '\*' in the third character place:

W2RN R xx\*

or W2MV R xx\*

or W2MR R xx\*

or W2RP R xx\*

An existence validation is performed in a library upper or equal to the target library for the called data elements only.

## DATA ELEMENTS

The indication of a parent Data Element code affects only the Data Element Definition in the source library. By default, a child Data Element remains attached to its parent. However, it is possible to suppress this link by entering the code '&&&&&&' in the parent Data Element field.

A child Data Element can be turned into a parent Data Element or may be assigned another parent by specifying a new parent Data Element code in the parent Data Element field.

In this case, the parent Data Element must be defined in a Library whose level is greater than or equal to the target Library.

A parent Data Element contained in a request must not have been previously processed as a source Data Element.

The format of the Data Element being moved remains the same, whatever the modification in relation to a parent Data Element.

If the target Data Element is used as an undefined Data Element, the format of its uses (on Segment or Report descriptions) must correspond to the format specified in the Definition.

The renaming of a key Data Element of a Data Structure (indicated as an argument on the Call of Data Structures '-CD' screen) is not allowed.

### PROGRAMS

Their processing goes through a check on libraries whose level is higher than or equal to that of the target library of the:

- called Macro-Structures,
- called Data Structures,
- Segments or Data Elements called in WORKING-STORAGE.

### SCREENS

Screens are processed individually. RMEN does not process the whole Dialogue. The Dialogue must therefore exist in a library whose level is higher than or equal to that of the target library.

### META-ENTITIES

A Meta-entity can be processed only if there is no other meta-entity bearing the same call code in the sub-network of the target Library.

**WARNING:** When the global upward move field contains 'ALL', an upward move of a Meta Entity involves the upward move of all of its User Entities contained in the source Library. If this field is blank, the User Entities remain in the source Library.

The existence of all Data Elements and User Relations called in the Definition lines is checked in a Library whose level is higher than or equal to the target Library.

### USER ENTITIES

The existence of the Meta-entity in a Library higher or equal to that of the target Library is checked, as is that of occurrences linked to the User Entities or details lines.

### MERISE ENTITIES



For MERISE Objects and Properties called in description lines, an existence check is performed in the library whose level is higher than or equal to that of the target library.

### DATABASE BLOCKS

The existence of MERISE objects or called segments is checked.

### VOLUMES

The existence of Reports called in the Volume Definition screen is checked.

### THE WORKSTATION ENTITIES

Calls of the `'//A'`, `'//M'`, `'//D'`, `'//O'` and `'//F'` type are used to extract all the WorkStation entities. The local entity type must be entered (in the entity type field) as well as the code of entity before processing, the code of the source Library and the code of the entity after processing.

The WorkStation methodology (MERISE, IFW, OMT, YSM...) is entered in a special field at position 10 in the command line corresponding to the environment (line type : 'E').

**WARNING:** One procedure execution can process entities related to only one Methodology.

## **Sub-Network and Entities Comparison**

### **CPSN - Introduction:**

The Sub-Network Comparison procedure (CPSN) compares the images of two sub-networks extracted by the PACX procedure (EXLI extractor, formatting for CPSN), which may or may not belong to the same network, or the images of entities extracted by the PACX procedure (EXTR or EXUE extractor, formatting for CPSN), in order to obtain the batch update transactions which will align the 'slave' sub-network or entities with the 'master' sub-network or entities.

- 'Master' sub-network = reference sub-network,
- 'Slave' sub-network = sub-network to align with the Master entity.
- 'Master' entity = reference entity,
- 'Slave' entity = entity to align with the reference entity

### EXECUTION CONDITION

None.

## ABNORMAL EXECUTIONS

If an abend occurs, the procedure can be restarted as it is once the problem has been solved.

### **CPSN - User Input:**

No specific line.

### NOTE

The sub-networks or entities to be compared must have been extracted via the PACX procedure (EXLI, EXTR or EXUE extractor; formatting for CPSN).

They must contain the same number of libraries (checked by the system) and have the same structure.

### **PACX - Description of Steps**

INPUT RECOGNITION: PTU001

INITIALIZATION OF KSDS WORKING FILE: IDCAMS

VSAM FILES CHECK: IDCAMS

EXTRACTION: PACX

This step extracts transactions according to user input.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database Index file
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database Data file
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database Extension Data
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GU	&INDSV..BVPGU	Input	Administration Database Users
PAC7PJ	&INDUN..&BASE.PJ(0)	Input	Archived transactions
PAC7MB	&&PACXMB	Input	User input

Code	Physical name	Type	Label
PAC7MA	DUMMY	Input	CPSN Master file
PAC7ES	DUMMY	Input	CPSN Slave file
PAC7BM	&&PACXBM	Input/Output	User input
PAC7MM	&&PACXMM	Input/Output	EXPU Work file
PAC7MJ	&&PACXMJ	Input/Output	EXPJ Work file
PAC7TE	&&PACXTE	Input/Output	RMEN Work file
PAC7RE	&&PACXRE	Input/Output	RMEN Work file
PAC7RM	&&PACXRM	Input/Output	RMEN Work file
PAC7WD	&&PACXWD	Input/Output	Extracted transactions
SYSEXT	&INDUV..SYSEXT.&USER	Input/Output	Work file
PAC7MV	&&MV	Output	Extracted transactions for UPDT
PAC7MR	&&MR	Output	Extracted transactions for REOR (EXPU)
PAC7MX	&&MX	Output	Non extracted entities (PACX)
PAC7GY	&&GY	Output	Extracted transactions for UPDP
PAC7TD	&&TD	Output	Extracted transactions for CPSN
PAC7UE	&&UE	Output	Extracted transactions for EXUE
PAC7IA		Report	General printout of the program stream
PAC7DD		Report	Errors on input transactions
PAC7ED		Report	Extractions report
PAC7EE		Report	Extractions report
PAC7EG		Report	Extractions report
PAC7EM		Report	Extractions report
PAC7EP		Report	Extractions report
PAC7EQ		Report	Extractions report
PAC7EU		Report	Extractions report
PAC7EZ		Report	Extractions report
SORTWK01		Sort	
SORTWK02		Sort	

Code	Physical name	Type	Label
SORTWK03		Sort	

Return codes:

- 0: No error
- 4: Error on user input (detailed in PAC7EE) or on the extractions for EXTR/EXUE (detailed in PAC7EZ)
- 8: Error on '\*' line (detailed in PAC7DD) or in EXLI (Database not available)

### DELETION OF THE KSDS WORKING FILE: IDCAMS

#### **PACX - Execution JCL**

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**          - EXTRACTIONS FROM DATABASE -
/**          - EXTRACTIONS COMPARATOR   -
/** -----
/**
/** THE PACX PROCEDURE ALLOWS TO PERFORM VARIOUS TYPES
/** OF DATA EXTRACTIONS FROM THE DEVELOPMENT DATABASE
/** VIA PAF EXTRACTOR.
/**
/** POSSIBLE VALUES FOR THE EXTRACTOR CODE INCLUDE:
/** - EXTR:  EXTRACTION OF ENTITIES
/** - EXTA:  EXTRACTION OF ENTITIES (EXTRACTED TRANSACTIONS
/**          ARE SORTED, ACCORDING TO THE INPUT
/**          IDENTIFICATION LINES ORDER.
/**          EACH REQUEST IS THUS PRECEDED BY A "*" LINE,
/**          EXTRACTED TRANSACTIONS WILL BE SORTED IN THE
/**          REQUEST ORDER).
/** - EXUE:  EXTRACTION OF USER ENTITIES
/** FOLLOWING VALUES ARE RESERVED FOR THE ADMINISTRATOR:
/** - EXLI:EXTRACTION OF LIBRARIES OR LIBRARY SUB-NETWORKS
/** - EXPJ:EXTRACTION OF JOURNAL (FORMATTING FOR CPSN IS
/**          NOT POSSIBLE)
/** - EXPU:EXTRACTION OF ENTITIES TO BE PURGED
/**          (FORMATTING FOR CPSN IS NOT POSSIBLE)
/** - RMEN:EXTRACTION OF ENTITIES FOR UPLOAD/REPLACEMENT/
/**          RECODING (FORMATTING FOR CPSN IS NOT POSSIBLE).
/**          RMEN IS SUBJECT TO A SEPARATE PURCHASE AGREEMENT
/** - CPSN:COMPARISON OF SUB-NETWORKS.
/**
/** -----
/**
/**/BVPPACX  PROC BASE=$BASE,          CODE OF VAPAC DATABASE
/**          INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
/**          INDSN='$INDSN',          INDEX OF SYSTEM NO VSAM FILES

```

```

//      INDUV='$INDUV',           INDEX OF USER VSAM FILES
//      INDUN='$INDUN',           INDEX OF USER NO VSAM FILES
//*:    VSAMCAT='$VCAT',           VSAM USER CATALOG
//*:    SYSTCAT='$SCAT',           VSAM SYSTEM CATALOG
//      PSBLIB='$PSBLIB',         LIBRARY OF PSBS
//      DBDLIB='$DBDLIB',         LIBRARY OF DBDS
//      RESLIB='$RESLIB',         IMS RESLIB
//      PROCLIB='$PRCLIB',        IMS PROCLIB
//      STEPLIB='$HLQ..SBVPMBR8', LOAD-MODULE LIBRARY
//      SORTLIB='$BIBT',          SORT LIBRARY
//      USER=,                     USER CODE
//      OUT='$OUT',                OUTPUT CLASS
//      OUTL='$OUT',               PRINT OUTPUT CLASS
//      UWK=$UWK,                  WORK UNIT
//      SPAMB='(TRK,(5,1),RLSE)', REQUEST FILE SPACE
//      SPAMV='(TRK,(50,10),RLSE)', SPACE OF UPDT FILE
//      SPAGY='(TRK,(50,10),RLSE)', SPACE OF UPDP FILE
//      SPATD='(TRK,(50,10),RLSE)', SPACE OF CPSN FILE
//      BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//      CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&PACXMB,DISP=(,PASS),
//      UNIT=&UWK,SPACE=&SPAMB,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//CARTE DD DDNAME=SYSIN
//*-----
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER'
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//      UNIT=&UWK,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//*-----
//MAXKEY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR

```

```

//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PACX EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPACX,PACX$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7PJ DD DSN=&INDUN..&BASE.PJ(0),DISP=SHR
//PAC7IA DD SYSOUT=&OUTL
//PAC7DD DD SYSOUT=&OUTL
//PAC7ED DD SYSOUT=&OUTL
//PAC7EE DD SYSOUT=&OUTL
//PAC7EG DD SYSOUT=&OUTL
//PAC7EM DD SYSOUT=&OUTL
//PAC7EP DD SYSOUT=&OUTL
//PAC7EQ DD SYSOUT=&OUTL
//PAC7EU DD SYSOUT=&OUTL
//PAC7EZ DD SYSOUT=&OUTL
//PAC7MA DD DUMMY

```

```

//PAC7ES DD DUMMY
//SYSEXT DD DSN=&INDUV. .SYSEXT.&USER,DISP=SHR
//PAC7MB DD DSN=&&PACXMB,DISP=(OLD,DELETE,DELETE)
//PAC7BM DD DSN=&&PACXBM,DISP=(,DELETE),UNIT=&UWK,
// DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7MM DD DSN=&&PACXMM,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//PAC7MJ DD DSN=&&PACXMJ,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=158,BLKSIZE=6320)
//PAC7TE DD DSN=&&PACXTE,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=323,BLKSIZE=6460)
//PAC7RE DD DSN=&&PACXRE,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=36,BLKSIZE=6012)
//PAC7RM DD DSN=&&PACXRM,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7WD DD DSN=&&PACXWD,DISP=(,DELETE),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7MV DD DSN=&&MV,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MR DD DSN=&&MR,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MX DD DSN=&&MX,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7TD DD DSN=&&TD,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPATD,
// DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7GY DD DSN=&&GY,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAGY,
// DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200)
//PAC7UE DD DSN=&&UE,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMV,
// DCB=(RECFM=FB,LRECL=230,BLKSIZE=6440)
//*-----
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER',COND=EVEN
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN. .BVPSY(DLSYSEXT),DISP=SHR
//PACROU DD DSN=&&DLSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----

```

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

---

## Session Management

### 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).

The utility consists in two procedures: the ESES preparation procedure and CSES compression procedure.

#### NOTE

The session 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 DSMS Journal (BJ), of the DSMS Database (BB), and of the Pactables Database (TC).

This utility includes two procedures: ESES and CSES.

## ESES - Session Numbers Extraction

### 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 (PACS with SAVE option)

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.

### **ESES - User Input**

A '\*' line with User code and Password is required.

One line (optional) per session number to force:

Position	Length	Value	Meaning
2	1	'S'	Line Code
3	4	nnnn	Original session number
7	4	nnnn	New session number

### **ESES - Description of Steps**

INPUT RECOGNITION: PTU001

CREATION OF THE SESSION MATCHING FILE: PTUESS

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7MB	&&ESESMB	Input	Input transactions
PAC7MV	&ESESFILE	Output	Session-number correspondence table
PAC7EU		Report	Extraction report
PAC7DD		Report	Batch procedure authorization option

Return codes:

- 8: No authorization for this procedure.

## ESES - Execution JCL

```

//* -----
//*      VISUALAGE PACBASE
//*
//* -----
//*      - SESSION NUMBERS CORRESPONDENCE TABLE -
//*
//* -----
//*
//* THE EXTRACTION OF SESSION NUMBERS PROCEDURE
//* (ESES) CREATES A CORRESPONDENCE-TABLE FILE LINKING
//* OLDER FROZEN SESSIONS AND NEW FROZEN SESSIONS.
//*
//* INPUT :
//* - USER IDENTIFICATION LINE (REQUIRED)
//* - COMMAND LINE :
//* COL 2  : "S"      LINE CODE
//* COL 3  : (4 N)   ORIGINAL SESSION NUMBER
//* COL 7  : (4 N)   NEW SESSION NUMBER
//* -----
//*
//BVPESES PROC BASE=$BASE,                CODE OF VAPAC DATABASE
//      INDSV='$INDSV',                    INDEX OF SYSTEM VSAM FILES
//      INDUV='$INDUV',                    INDEX OF VSAM USER FILES
//      INDSN='$INDSN',                    INDEX OF SYSTEM NON-VSAM FILES
//*:     VSAMCAT='$VCAT',                   USER VSAM CATALOG
//*:     SYSTCAT='$SCAT',                   SYSTEM VSAM CATALOG
//      ESESEFILE=,                        SESSION NUMBER FILE DSNAME
//      VOLS='SER=$VOLUN',                  SESSION NUMBER FILE VOLUME
//      UNITS=$UNITUN,                     SESSION NUMBER FILE UNIT
//      STEPLIB='$HLQ..SBVPMBR8',          LIBRARY OF LOAD-MODULES
//      SPAMB='(TRK,(30,10))',            SPACE OF EXTRACTED TRANSACTIONS
//      UWK=$UWK,                          WORK UNIT
//      OUT=$OUT,                          OUTPUT CLASS
//      PSBLIB='$PSBLIB',                  LIBRARY OF PSB'S
//      DBDLIB='$DBDLIB',                  LIBRARY OF DBD'S
//      RESLIB='$RESLIB',                  IMS RESLIB
//      PROCLIB='$PRCLIB',                 IMS PROCLIB
//      BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//      CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&ESESMB,DISP=(,PASS),
//      UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//-----
//PTUESS EXEC PGM=DFSRR000,REGION=$REGSIZ,
//      PARM=(DLI,BVPTUESS,PTUESS$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR

```

```

//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7MB   DD DSN=&&ESESMB,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&ESESFILE,DISP=(,CATLG),
//          UNIT=&UNITS,VOL=&VOLS,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,BLKSIZE=5000,LRECL=5)
//*
```

## CSES - Compression of Session Numbers

### CSES - Introduction

The Compression of Session Numbers procedure (CSES) compresses the session numbers of the development 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.

Yet, all the backups to be processed must be valid.

### CSES - User Input

A \* line with User Code and Password.

The user input is used to indicate the list of files to be retrieved (PC, PJ, BB, BJ, and TC), in order to execute the retrieval after one or several runs.

The line is built as follows:

Position	Length	Value	Meaning
2	1	'S'	Line code
3	21		Code of the files to retrieve (PC PJ BB BJ TC) separated with a blank
33	4		If the DSMS database has to be retrieved: development Database logical code

### CSES - Description of Steps

INPUT RECOGNITION: PTU001

COMPRESSION OF SESSION NUMBERS: PTUCSS

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7MV	&ESESFILE	Input	Session numbers correspondence table
PAC7MB	&&CSESMB	Input	Parameter line
PAC7PC	&INDUN..&BASE.PC(0)	Input	Development Database backup
PAC7PD	&INDUN..&BASE.PD(0)	Input	If Dispatch option of the backup: Backup 2 of the development Database
PAC7PY	&INDUN..&BASE.PY(0)	Input	If Dispatch option of the backup: Backup 3 of the development Database
PAC7CP	&INDUN..&BASE.PC(+1)	Output	If Dispatch option of the backup:
PAC7DP	&INDUN..&BASE.PD(+1)	Output	If Dispatch option of the backup: Backup 2 of the development Database

Code	Physical name	Type	Label
PAC7YP	&INDUN..&BASE.PY(+1)	Output	If Dispatch option of the backup: Backup 3 of the development Database
PAC7PJ	&INDUN..&BASE.PJ(0)	Input	Backup of the development Database Journal
PAC7JP	&INDUN..&BASE.PJ(+1)	Output	Backup of the development Database Journal
PACDBB	&DSMS.BB(0)	Input	DSMS Database backup (if DSMS is installed)
PACDJB	&DSMS.BB(+1)	Output	DSMS Database backup (if DSMS is installed)
PACDDJ	&DSMS.BJ(0)	Input	Retrieval of DSMS archived Journal (if DSMS is installed)
PAC7JD	&DSMS.BJ(+1)	Output	Retrieval of DSMS archived Journal (if DSMS is installed)
PACDTC	&DSNTC0	Input	Retrieval of Pactables backup (if Pactables is installed)
PACDCT	&DSNTC1	Output	Retrieval of Pactables backup (if Pactables is installed)
PAC7EU		Report	Execution output
PAC7DD		Report	Batch procedure authorization option

Return codes:

- 8: No authorization for this procedure.

### CSES - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**          - COMPRESSION OF SESSION NUMBERS -
/**
/** -----
/**
/** THE COMPRESSION OF SESSION NUMBERS PROCEDURE (CSES)
/** COMPRESSES THE SESSION NUMBERS OF THE DEVELOPMENT
/** DATABASE LOGICAL BACKUPS, THE PACTABLES DATABASE IF
/** THIS MODULE IS INSTALLED ON THE SITE, AND THE DSMS DATA
/** BASE IF THIS MODULE IS INSTALLED ON THE SITE. IT USES
/** THE CORRESPONDENCE TABLE CREATED BY THE ESES PROCEDURE.
/** THE RESULTING FILES MUST BE RESTORED.

```

```

/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/** - COMMAND LINE :
/** COL 2 : "S" LINE CODE
/** COL 3 : (21 CAR.) CODE OF THE FILES TO RETRIEVE (PC
/** PJ PG PP BB BJ TC) SEPARATED WITH A BLANK
/** COL 33 : (4 CAR.) IF THE DSMS DATABASE HAS TO BE
/** RETRIEVED : DEVELOPMENT DATABASE LOGICAL CODE
/** -----
/**
//BVPCSES PROC BASE=$BASE, CODE OF VAPAC DATABASE
// INDSV='$INDSV', INDEX OF SYSTEM VSAM FILES
// INDUN='$INDUN', INDEX OF VA PAC BACKUP FILES
// INDSN='$INDSN', INDEX OF SYSTEM NON-VSAM FILES
// DSMSBB0=, DSMS DATABASE BACKUP (0)
// DSMSBB1=, DSMS DATABASE BACKUP (+1)
// DSMSBJ0=, RETRIEVAL OF DSMS ARCHIVE (0)
// DSMSBJ1=, RETRIEVAL OF DSMS ARCHIVE (+1)
/**: VSAMCAT='$VCAT', USER VSAM CATALOG
/**: SYSTCAT='$SCAT', SYSTEM VSAM CATALOG
// ESEFILE=, SESSION NUMBERS FILE DSNAME
// DSNTC0=, TABLES BACKUP FILE DSNAME (0)
// DSNTC1=, TABLES BACKUP FILE DSNAME (+1)
// STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
// UWK=$UWK, WORK UNIT
// DSCB='$DSCB', DSCB MODEL FILE
// OUT=$OUT, OUTPUT CLASS
// VOLS='SER=$VOLUN', BACKUP VOLUME
// UNITS=$UNITUN, BACKUP UNIT
// SPAPC='(TRK,(300,10),RLSE)', PC BACKUP SPACE
// SPAPD='(TRK,(300,10),RLSE)', PD BACKUP SPACE
// SPAPJ='(TRK,(100,10),RLSE)', PJ BACKUP SPACE
// SPAPY='(TRK,(100,10),RLSE)', PY BACKUP SPACE
// PSBLIB='$PSBLIB', LIBRARY OF PSB'S
// DBDLIB='$DBDLIB', LIBRARY OF DBD'S
// RESLIB='$RESLIB', IMS RESLIB
// PROCLIB='$PRCLIB', IMS PROCLIB
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&CSESMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//-----
//PTUCSS EXEC PGM=DFSRRC00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUCSS,PTUCSS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR

```

```

//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7EU   DD SYSOUT=&OUT
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7MB   DD DSN=&&CSESEMB,DISP=(OLD,PASS)
//PAC7MV   DD DSN=&ESESFILE,DISP=SHR
//PAC7PC   DD DSN=&INDUN..&BASE.PC(0),DISP=SHR
//PAC7PD   DD DSN=&INDUN..&BASE.PD(0),DISP=SHR
//PAC7PY   DD DSN=&INDUN..&BASE.PY(0),DISP=SHR
//PAC7CP   DD DSN=&INDUN..&BASE.PC(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,
//          SPACE=&SPAPC,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=27998,LRECL=1023)
//PAC7DP   DD DSN=&INDUN..&BASE.PD(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,
//          SPACE=&SPAPD,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=27998,LRECL=1023)
//PAC7YP   DD DSN=&INDUN..&BASE.PY(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,
//          SPACE=&SPAPY,
//          DCB=(&DSCB,RECFM=VB,BLKSIZE=27998,LRECL=1023)
//PAC7PJ   DD DSN=&INDUN..&BASE.PJ(0),DISP=SHR
//PAC7JP   DD DSN=&INDUN..&BASE.PJ(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,
//          SPACE=&SPAPJ,
//          DCB=(&DSCB,RECFM=FB,BLKSIZE=27880,LRECL=170)
//PACDBB   DD DSN=&DSMSBBO,DISP=SHR
//PACDJB   DD DSN=&DSMSBB1
//PACDDJ   DD DSN=&DSMSBJ0,DISP=SHR

```

```
//PACDJD DD DSN=&DSMSBJ1
//PAC7TC DD DSN=&DSNTCO,DISP=SHR
//PAC7CT DD DSN=&DSNTC1
//*
```

---

## Storage Optimization of Multi-Volumes

### 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.

The STOP procedure is used to allocate data on several volumes. The successive data blocks are alternately distributed on each volume.

### EXECUTION CONDITIONS

None, since the database is not updated directly.

### STOP - User Input

A '\*' line with user code and password.

One command line:

Position	Length	Value	Meaning
2	1	' '	Blank line code
3	8	nnnnnnnn	Number of data (including gaps) See back-up statistics on input to the procedure
11	8	gggggggg	Number of disks
19	8	bbbbbbbb	Number of records per data block

### CALCULATION OF THE NUMBER OF RECORDS PER BLOCK

Let N be the number of records per physical block.

Use a multiple of N.

Example:



Let's use a physical block of 4,096K

Let's use the AR file (length of records=144)

In this case, N=27.

Let's use a block of 100 physical block.

The number of records par block is therefore 2700.

With nnnnnnn given to be divided among 3 volumes, the command line looks like the following:

\_\_nnnnnnn\_\_\_\_\_3\_\_\_\_2700

### PRINTED REPORT

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.

## **STOP - Description of Steps**

INPUT RECOGNITION: PTU001

STORAGE OPTIMIZATION OF DATA: PTUR00

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
PAC7PC	&INDUN..&BASE.PC(0)	Output	Sequential image of the development Database
PAC7PD	&INDUN..&BASE.PD(0)	Output	If Dispatch option of the backup: Sequential image 2 of the development Database

Code	Physical name	Type	Label
PAC7PY	&INDUN..&BASE.PY(0)	Output	If Dispatch of the backup: Sequential image 3 of the development Database
PAC7MB	&&STOPMB	Input	User transaction
PAC7CP	&INDUN..&BASE.PC(+1)	Output	Sequential image of the Database
PAC7DP	&INDUN..&BASE.PD(+1)	Output	If Dispatch option of the backup: Sequential image 2 of the Database
PAC7YP	&INDUN..&BASE.PY(+1)	Output	If Dispatch option of the backup: Sequential image 3 of the Database
PAC7EU		Report	Report
PAC7DD		Report	Batch procedure authorization option
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

## STOP - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/** - STORAGE OPTIMIZATION OF DEVELOPMENT DATABASE DATA -
/**      (FOR MULTI VOLUMES AR FILE)
/** -----
/**
/** 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.
/**
/** INPUT :
/** COL 2  : " "   BLANK LINE CODE
/** COL 3  : (8 N) NUMBER OF DATA (INCLUDING GAPS)
/**          SEE BACK-UP STATISTICS ON INPUT
/**          THE PROCEDURE
/** COL 11 : (8 N) NUMBER OF DISKS
/** COL 19 : (8 N) NUMBER OF RECORDS PER DATA BLOCK
/** -----
/**
/**BVPSTOP PROC BASE=$BASE,          CODE OF VAPAC DATABASE
/**          INDUN='$INDUN',          INDEX OF USER NON VSAM FILES
/**          INDSV='$INDSV',          INDEX OF SYSTEM VSAM FILES
/**          INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
/**          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
/***:          VSAMCAT='$VCAT',       USER VSAM CATALOG
/***:          SYSTCAT='$SCAT',       SYSTEM VSAM CATALOG

```

```

//          OUT=$OUT,                OUTPUT CLASS
//          SORTLIB='$BIBT',         SORT LIBRARY
//          PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//          RESLIB='$RESLIB',        IMS RESLIB
//          PROCLIB='$PRCLIB',       IMS PROCLIB
//          CYL=10,                  SIZE OF WORK FILE
//          UWK='$UWK',              SIZE OF WORK FILE
//          VOLS='SER=$VOLUN',       BATABASE BACKUP VOLUME
//          UNITS=$UNITUN,           BACKUP UNIT (DISK OR CARTRIDGE)
//          VOLS2='SER=$VOLUN',     BATABASE BACKUP 2 VOLUME
//          UNITS2=$UNITUN,         BACKUP 2 UNIT (DISK OR CARTRIDGE)
//          DSCB='$DSCB',           DSCB MODEL FILE
//          SPAPC='(TRK,(300,10),RLSE)',  BACKUP 'SPACE'
//          SPAPD='(TRK,(300,10),RLSE)',  BACKUP 'SPACE'
//          SPAPY='(TRK,(300,10),RLSE)',  BACKUP 'SPACE'
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****-----*
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&STOPMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*-----*
//PTUR00 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUR00,PTUR00$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDRR DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)

```

```

//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7MB DD DSN=&&STOPMB,DISP=(OLD,DELETE)
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7PC DD DSN=&INDUN..&BASE.PC(0),DISP=SHR
//PAC7PD DD DSN=&INDUN..&BASE.PD(0),DISP=SHR
//PAC7PY DD DSN=&INDUN..&BASE.PY(0),DISP=SHR
//PAC7CP DD DSN=&INDUN..&BASE.PC(+1),DISP=(,CATLG,DELETE),
//
// UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPC,
//
// DCB=(&DSCB,RECFM=VB,BLKSIZE=27998,LRECL=1023)
//PAC7DP DD DSN=&INDUN..&BASE.PD(+1),DISP=(,CATLG,DELETE),
//
// UNIT=&UNITS2,VOL=&VOLS2,SPACE=&SPAPD,
//
// DCB=(&DSCB,RECFM=VB,BLKSIZE=27998,LRECL=1023)
//PAC7YP DD DSN=&INDUN..&BASE.PY(+1),DISP=(,CATLG,DELETE),
//
// UNIT=&UNITS2,VOL=&VOLS2,SPACE=&SPAPY,
//
// DCB=(&DSCB,RECFM=VB,BLKSIZE=27998,LRECL=1023)
//PAC7EU DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//*
```

---

## Database statistics

### STAT - Introduction

This procedure creates, from the files output by the save procedure, a sequential file which can be used to calculate statistics.

#### EXECUTION CONDITIONS

None.

#### OUTPUT RESULT

This procedure outputs a sequential file whose records are 100 character-long and whose contents are:

- a global record:

Pos.	Len.	Format/Value	Meaning
9	10	'GLOBAL '	Constant
19	3	9(3)	Number of libraries
22	8	9(8)	Index: number of records
30	8	9(8)	Index number of deleted records
38	8	9(8)	Data: number of records (gaps + data)

Pos.	Len.	Format/Value	Meaning
46	8	9(8)	Data: number of deleted records
54	8	9(8)	Data: number of 'gap' records
62	8	9(8)	Data: number of 'data' records
70	8	9(8)	Data: number of records in frozen sessions
78	8	9(8)	Data: number of records in current sessions
86	8	9(8)	Container: number of records

- One or more records per library which has lower-level libraries:

Pos.	Len.	Format/Value	Meaning
1	3	X(3)	Library code
9	6	'LIBINF'	Constant (BIBINF for a French database)
16	2	X(2)	First record: ' ' Next records: 01, 02, 03, etc ...
19	81		Table containing 27 items (3-character long) which represent lower-level libraries

- One or more records per library which has higher-level libraries:

Pos.	Len.	Format/Value	Meaning
1	3	X(3)	Library code
9	6	'LIBSUP'	Constant (BIBSUP for a French database)
16	2	X(2)	First record: ' ' Next records: 01, 02, 03, etc ...
19	81		Table containing 27 items (3-character long) which represent higher-level libraries

- One record per library/session/line type:

Pos.	Len.	Format/Value	Meaning
1	3	X(3)	Library code
4	4	9(4)	Session number
8	1	X(1)	Session type
9	10	X(10)	Line type
19	8	9(8)	Number of data items
27	8	9(8)	Number of deleted data items

The line type corresponds to the PAF table code. The following codes have been added:

- \$ttDSC/YttDSC: User Entity description, tt being the call type (each description type is not detailed),
- xxxLOCKS: locks and timestamps, xxx being the entity type: TXT, DEL,... (xxxBLOCAGE for a French Database),
- xxxKWD: keywords, xxx being the entity type: TXT, DEL, ... (xxxMCL for a French Database),
- LONGV3: Long data attached to comments (-GC), generation elements (-GG), generation options (-GO) and error messages lines (-GE),
- LONG4: Long data attached to the layout lines of Reports,
- LONGY3: Long data attached to the User Entity Definition,
- LONGY4: Long data attached to the User Entity Descriptions.

### STAT - User input

A '\*' line with the user code and password.

### STAT - Description of steps

INPUT RECOGNITION: PTU001

FORMATTING OF THE SEQUENTIAL FILE: PTUSTA

Code	Physical Name	Type	Name
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the Development Database
PAC7PD	&INDUN..&BASE.PD(0)	Input	If Dispatch option in the save procedure: Sequential image 2 of the network
PAC7PY	&INDUN..&BASE.PY(0)	Input	If Dispatch option in the save procedure: Sequential image 3 of the network
PAC7MB	&&STATMB	Input	User transactions
PAC7ST	&&PAC7ST	Output	Output file (length= 100)
PAC7DD		Report	Execution report
SORTWK01		Sort	
SORTWK02		Sort	

Code	Physical Name	Type	Name
SORTWK03		Sort	

## STAT - Execution JCL

```

//* -----
//*      VISUALAGE PACBASE
//*
//* -----
//*      GENERATE DATABASE STATISTICS
//*
//* -----
//*
//BVPSTAT  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          INDSN='$INDSN',                INDEX OF SYSTEM NON VSAM FILES
//          INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
//          INDUV='$INDUV',                INDEX OF USER VSAM FILES
//          INDUN='$INDUN',                INDEX OF USER NON VSAM FILES
//*:       VSAMCAT='$VCAT',                USER VSAM CATALOG
//*:       SYSCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',              SORT LIBRARY
//          CYL=(10,10)',                SORTWORK SPACE
//          OUT=$OUT,                    OUTPUT CLASS
//          OUTL=$OUT,                   OUTPUT CLASS OF REPORTS
//          UWK=$UWK,                    WORK UNIT
//          SPAST=(TRK,(10,5))',         EXTRACTED FILE SPACE
//          PSBLIB='$PSBLIB',            LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//          RESLIB='$RESLIB',            IMS RESLIB
//          PROCLIB='$PRCLIB',           IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&STATMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR

```

```

//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//*-----
//PTUSTA  EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUSTA,PTUSTA$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS     DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSOUX  DD SYSOUT=&OUT
//DDSNAP  DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON  DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE  DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN  DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR  DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU  DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7PC  DD DSN=&INDUN..&BASE.PC(0),DISP=OLD
//PAC7PD  DD DSN=&INDUN..&BASE.PD(0),DISP=SHR
//PAC7PY  DD DSN=&INDUN..&BASE.PY(0),DISP=SHR
//PAC7MB  DD DSN=&&STATMB,DISP=(OLD,DELETE)
//PAC7ST  DD DSN=&&PAC7ST,DISP=(,PASS),UNIT=&UWK,SPACE=&SPAST,
//          DCB=(RECFM=FB,LRECL=100,BLKSIZE=3400)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7DD  DD SYSOUT=&OUTL
//

```



---

## Chapter 5. Analysis of Activity and Quality Control

---

### Analysis of Activity

#### ACTI - Introduction

The ACTI procedure is an optional utility, and its use depends on the corresponding purchase agreement.

The Specifications Dictionary manages all the data related to the various applications being developed or maintained at the site.

The Journal file contains all the database update transactions. As such, it reflects user activity.

With the Journal Statistics Utility (ACTI), this activity can be monitored and presented in the form of charts.

The Journal Statistics Utility allows the Database Manager to query the Journal backup file based on various parameters:

- library code,
- user code,
- entity type,
- entity code,
- line code,
- transaction type,
- date of update,
- time of update,
- session number of update,
- transaction code,
- change number.

These criteria are used to specify the REQUEST AREA.

Results are obtained in the form of three types of charts, i.e., statistical reports, curve-type graphs, or lists of transactions.

This output will be printed according to the selected PAGE LAYOUT. Statistics and graphs are sorted and calculated according to the user request.

- Output Report Type,

- page layout criteria,
- Request Area,
- Data sequencing mode,
- Activity calculation mode.

### EXECUTION CONDITIONS

None.

## **ACTI - Query Language**

### REQUEST CODING

A Journal Statistics Request consists of five different types of lines, identified by the following KEYWORDS:

- Output : Output Report Type,
- Page : Page Layout (page breaks),
- Area : Request Area,
- Line : Statistical Report Lines,
- Column : Statistical Report Columns,
- Abscissa : Curve-type graph Abscissas,
- Ordinate : Curve-type graph Ordinates.

The meaning of the keywords, the parameters which define them, as well as their compatibility are explained in the 'Keywords Meaning and filling modes' paragraph.

The Output line is required; the Page and Area lines are optional. The Line, Column, Abscissa, and Ordinate lines are either required or prohibited, depending on the requested output report type.

Only the first three characters of a keyword are used to identify a line type.

On the printed report, each request line is explicitly stated on the first page and an explicit error message is generated in case of a rejected line.

Request lines must be entered in the following order:

Output Page Area Line Column Abscissa Ordinate

Any error in this sequence will be considered as the beginning of another request.

The user may enter up to 10 requests at the same time.

The purpose of the ':' character is to mark the end of the keyword.

The rest of the line contains the parameters of each characteristic.

### PARAMETERS

Parameters are used to define page layouts, lines and abscissas. These are called 'Presentation Criteria'.

Parameters followed by '=' and a value are called 'Selection Criteria'.

Parameters which define calculations are called 'Calculations'.

The coding, meaning and compatibility of the parameters are described in the 'Parameters Definition and Comments' paragraph.

### SEPARATORS

The data entered on request lines are separated and grouped together using the following characters:

- ':' = end of keyword,
- '=' = link between a parameter and its value,
- '(' )' = set of parameters for calculations,
- ',' = parameter or calculation separator,
- '/' = calculation combination,
- '\*' = generic selection,
- 'Blank' = end of line (subsequent data is entered for documentary purposes).

### KEYWORDS: MEANING AND FILLING MODES

OUT(put) : Output report type

This type of line is required at the beginning of each request.

The parameters used to define the output report type are:

- STA for statistics
- GRA for graph
- LIS for list

PAG(es) : Page layout

This type of line is used to indicate at which level a page skip is to be inserted.

The Page layout line is optional.

Headings are printed for each level, as well as totals for the statistical reports.

The page layout is defined by a series of parameters (three maximum separated by the ',' character) identifying data from the Journal, and called 'presentation criteria'.

Example: A page skip may be requested for each user and for each library.

ARE(a) : Request area

This type of line is used to define the transactions to be taken into account.

The REQUEST AREA line is optional.

The Request Area is defined by parameters (separated by the ',' character) followed by the '=' character and the selected value.

Example: The request applies to only some users and for a given period of time.

LIN(es) : Data sorting mode

or

ABS(cissa)

This type of line is used to define either the lines of a statistical report or the X-axis of a curve-type graph.

It is required for both statistical reports and curve-type graphs. However, it is not permitted for transaction lists.

There may be several lines of this type for statistical report.

The Data Sorting Mode may be defined by Presentation Criteria, as well as Selection Criteria. Parameters and values are separated by the ',' character.

Example: Data is sorted by entity type for a statistical report, or by week for a curve-type graph.

COL(umns) : Activity calculation mode

or

ORD(inate)

This type of line defines the columns of a statistical report or the ordinates of a curve-type graph (maximum of seven columns or curves).

It is required for both statistical reports and curve-type graphs. However, it is not permitted for transaction lists.

Each column or curve is determined by a calculation, followed by bracketed Selection Criteria. Columns or curves, parameters and values, are all separated by the '/' character.

A printing character (&Cn\*dofHAR='X') must be specified for each curve.

A statistical report column may be defined by the relationship between two calculations; these calculations are separated by the '/' character.

Example: A first column or a first curve may be a calculation of the transactions entered on-line, while a second one may show the ratio between the input transactions and the real transactions.

#### PARAMETERS: DEFINITION AND COMMENTS

&LIB : Library code

This parameter is used as a Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

A generic selection may be requested by simply replacing every appropriate character by the '\*' character.

&USER : User code

This parameter is used as a Presentation and Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

A generic selection may be requested by simply replacing every appropriate character by the '\*' character.

&ENTG : Entity type

This parameter is used as a Presentation and Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

&ENTD : Line code / Entity type

This parameter is used as a Presentation and Selection Criterion to define the Data Sorting Mode.

Values are selected according to the entity type entered in the preceding parameter.

&LICO : Line code

This parameter is used as a Presentation and Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and Activity Calculation Mode.

Values are selected according to the batch line codes.

&ENT : Entity code

This parameter is used as a Presentation and Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

A generic selection may be requested by simply replacing every appropriate character by the '\*' character.

Values are selected according to the entity type and code.

&INPT : Input type

This parameter is used as a Presentation and Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

The value 'B' corresponds to batch input mode; any other value corresponds to on-line input mode.

&D1 : Starting date

This parameter is used as a Selection Criterion to define the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

This parameter has to be followed by a date (MMDDCCYY). If this parameter is missing, the starting date coincides with the beginning of the Journal.

&D2 : End date

This parameter is used as a Selection Criterion to define the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

This parameter has to be followed by a MMDDCCYY date format.

If this parameter is missing, the end date coincides with the end of the Journal.

&S1 : Starting session

This parameter is used as a Selection Criterion to define the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

This parameter has to be followed by a five-character field: a four-character session field and one-character session status field.

If this parameter is missing, the starting session coincides with the beginning of the Journal.

&S2 : Final session

This parameter is used as a Selection Criterion to define the Request Area, the Data Sorting Mode, and the Activity Calculation mode.

This parameter has to be followed by a five-character session field: a four-character session number and a one-character session status. If this parameter is missing, the final session coincides with the end of the Journal.

&DAY : Day-by-day presentation

Used as a Presentation Criterion to define the page layout and the data sorting mode.

To define an X-axis, this parameter must be followed by the '=' character and the number of characters corresponding to the curve step (its default value is one character).

&WEEK : Week-by-week presentation

Used as a presentation criterion to define the page layout and the data sorting mode.

To define an X-axis, this parameter must be followed by the '=' character and the number of characters corresponding to the curve step (its default value is one character).

&MON : Month-by-month presentation

Used as a presentation criterion to define the page layout and the data sorting mode.

To define an X-axis, this parameter must be followed by the '=' character and the number of characters corresponding to the curve step (its default value is one character).

&YEAR : Year-by-year presentation

Used as a presentation criterion to define the page layout and the data sorting mode.

To define an X-axis, this parameter must be followed by the '=' character and the number of characters corresponding to the curve step (its default value is one character).

&SESS : Presentation by session

Used as a presentation criterion to define the page layout and the data sorting mode.

The user cannot use it to select sessions (the '=' character is therefore unnecessary).

&CHAR : Printing curve character

May only be used to define the activity calculation mode relative to the curve-type graphs.

It must follow (within parentheses) the calculation defining a curve.

&INTR : Number of input transactions

May only be used to define the activity calculation mode. Each Journal transaction is an input transaction.

&RETR : Number of real transactions

May only be used to define the activity calculation mode.



A Journal transaction is effective, provided it is not modified by another transaction and it is not itself a deletion transaction. This concept is linked to the presentation criteria, i.e. a transaction which is modified once a day is effective every day with a day-by-day presentation; it is effective only once with another presentation.

&H1 : Starting hour

This parameter is used as a Selection Criterion to define the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

This parameter has to be followed by a HHMMSS hour format.

If this parameter is missing, the starting hour coincides with the beginning of the Journal.

&H2 : End hour

This parameter is used as a Selection Criterion to define the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

This parameter has to be followed by a HHMMSS hour format.

If this parameter is missing, the end hour coincides with the beginning of the Journal.

&MIN : Minute-by-minute presentation

Used as a presentation criterion to define the page layout and the data sorting mode.

To define an X-axis, this parameter must be followed by a '=' character and the number of characters corresponding to the curve step (its default value is one character)

&HOUR : Hour-by-hour presentation

Used as a presentation criterion to define the page layout and the data sorting mode.

To define an X-axis, this parameter must be followed by a '=' character and the number of characters corresponding to the curve step (its default value is one character)

&MCOD : Transaction code

This parameter is used as a Presentation and Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

&DSMS : Change number

This parameter is used as a Presentation and Selection Criterion to define the Page Layout, the Request Area, the Data Sorting Mode, and the Activity Calculation Mode.

Parameter	AREa	PAGes	OUTput	OUTput
			STA	GRA
			LIN COL	ABS ORD
&LIB	YES	YES	YES	YES
&USER	YES	YES	YES	YES
&ENTG	YES	YES	YES	YES
&ENTD		YES	YES	
&LICO	YES	YES	YES	YES
&ENT	YES	YES	YES	YES
&INPT	YES	YES	YES	YES
&D1=				
MMDDCCYY	YES		YES	YES
&D2=				
MMDDCCYY	YES		YES	YES
&S1=9999Z	YES		YES	YES
&S2=9999Z	YES		YES	YES
&MIN	YES	YES	YES	=
&HOUR	YES	YES	YES	=
&DAY	YES	YES	YES	=
&WEEK	YES	YES	YES	=
&MON	YES	YES	YES	=
&YEAR	YES	YES	YES	=
&SESS		YES	YES	
&MCOd		YES	YES	YES
&DSMS		YES	YES	YES
&CHAR				CALCULATION
&INTR				CALCULATION

Parameter	AREa	PAGes	OUTput	OUTput
&RETR				CALCULATION

'=' : the parameter must be followed by the separator character '=' and the curve step;

CALCULATION : only used in the Activity Calculation Mode.

### ACTI - User Input

A '\*' line with user code and password.

Specific input needed for this procedure is described in the optional utilities Reference Manual, in the chapter dedicated to this procedure.

### ACTI - Description of Steps

INPUT RECOGNITION: PTU001

EXTRACTION: PTU630

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Output	Development Database data
B7AN\$BASE	&INDUV..&BASE.AN	Output	Development Database index
PAC7PJ	&INDUN..&BASE.PJ(0)	Input	Archived update transactions
PAC7MB	&&ACTIMB	Input	Update transactions
PAC7ST	&&PAC7ST	Output	Selected reports transactions (length=247)
PAC7DD		Report	Batch procedure authorization option

Return codes:

- 0: OK
- 8: No authorization on batch procedures.
- 12: System error.

## PRINTING OF RESULTS: PTU640

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7ST	&&PAC7ST	Input	Transactions for selected reports
PAC7IV		Report	Selected reports
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### ACTI - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**           - ACTIVITY ANALYSIS -
/**
/** -----
/**
/** THE JOURNAL FILE CONTAINS ALL THE DATABASE UPDATE
/** TRANSACTIONS. AS SUCH, IT REFLECTS USER ACTIVITY.WITH
/** THE JOURNAL STATISTICS UTILITY (ACTI), THIS ACTIVITY
/** CAN BE MONITORED AND PRESENTED IN THE FORM OF CHARTS.
/** THE JOURNAL STATISTICS UTILITY ALLOWS THE DATABASE
/** MANAGER TO QUERY THE JOURNAL BACKUP FILE BASED ON
/** VARIOUS PARAMETERS:
/**   - LIBRARY CODE
/**   - USER CODE
/**   - ENTITY TYPE
/**   - ENTITY CODE
/**   - LINE CODE
/**   - TRANSACTION TYPE (C,M,D)
/**   - DATE OF UPDATE
/**   - SESSION NUMBER OF UPDATE
/** -----
/**
/** THE JOURNAL FILE CONTAINS ALL THE DATABASE UPDATE
/** TRANSACTIONS. AS SUCH, IT REFLECTS USER ACTIVITY.WITH
/** THE JOURNAL STATISTICS UTILITY (ACTI), THIS ACTIVITY
/** CAN BE MONITORED AND PRESENTED IN THE FORM OF CHARTS.
/** THE JOURNAL STATISTICS UTILITY ALLOWS THE DATABASE
/** MANAGER TO QUERY THE JOURNAL BACKUP FILE BASED ON
/** VARIOUS PARAMETERS:
/**   - LIBRARY CODE
/**   - USER CODE
/**   - ENTITY TYPE
/**   - ENTITY CODE
/**   - LINE CODE
/**   - TRANSACTION TYPE (C,M,D)
/**   - DATE OF UPDATE

```

```

//*      - SESSION NUMBER OF UPDATE
//* -----
//*
//BVPACTI  PROC BASE=$BASE,           CODE OF VAPAC DATABASE
//          INDSV='$INDSV',           INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',           INDEX OF SYSTEM NON-VSAM FILES
//          INDUN='$INDUN',           INDEX OF USER NON-VSAM FILES
//          INDUV='$INDUV',           INDEX OF USER VSAM FILES
//*:       SYSTCAT='$SCAT',           SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8', LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',          SORT LIBRARY
//          OUT=$OUT,                  OUTPUT CLASS
//          SPAMV='(TRK,(30,10),RLSE)', EXTRACTED TRANSAC. SPACE
//          CYL=5,                      SORTWORK SIZE
//          UWK=$UWK,                   WORK UNIT
//          PSBLIB='$PSBLIB',          LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',          LIBRARY OF DBD'S
//          RESLIB='$RESLIB',           IMS RESLIB
//          PROCLIB='$PRCLIB',         IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&ACTIMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PTU630   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU630,PTU630$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR

```

```

//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7MB   DD DSN=&&ACTIMB,DISP=(OLD,PASS)
//PAC7PJ   DD DSN=&INDUN..&BASE.PJ(0),DISP=SHR
//PAC7ST   DD DSN=&&PAC7ST,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=247,BLKSIZE=7657),
//          SPACE=&SPAMV
//PAC7DD   DD SYSOUT=&OUT
//*-----
//PTU640   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU640,PTU640$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=(00,NE,PTU630)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)

```

```
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7ST DD DSN=&&PAC7ST,DISP=(OLD,PASS)
//PAC7IV DD SYSOUT=&OUT
//*
```

---

## Pacbench Quality Control

### Introduction

The Pacbench Quality Control (PQC) facility is optional, and its use depends on the corresponding purchase agreement.

The Pacbench Quality Control facility is divided into two components:

- The Analysis component, to evaluate the quality of applications in use. This is based either on standard rules or on rules customized by the purchaser site,
- The Quality rule extraction component, customized by the purchaser site.

Two purchase options are therefore available:

- A basic option providing standard rules for quality control;
- A quality rule CUSTOMIZATION option.

The components supplied on the installation tape are:

- For both purchase options:
  - A Batch Quality Analysis procedure (PQCA);
  - A set of 'compiled' standard quality rules, in the form of a sequential file (see the Environment & Installation manual).
- For the CUSTOMIZATION option:
  - A batch procedure for the extraction and 'compilation' of the customized rules (PQCE);
  - A data element dictionary and the user entity needed for the customization of the rules, in the form of Batch transactions that the user enters in his/her own dictionary via a Batch update (UPDT). (See the Environment & Installation manual.)

### Analysis

#### **PQCA - Introduction**

The PQCA procedure carries out an analysis of the quality of the applications, according to either standard rules or user-defined rules.

#### CHARACTERISTICS

The procedure invokes a unique program (BVPACQ), which is used as a base for links to the various programs used by the procedure.

All the programs called during the procedure are therefore considered to be sub-programs of BVPACQ, with which they communicate via a Communication Area and special return codes.

It is functionally identical to the GPRT procedure.

The procedure is split up into 'sub-chains', identified by a 1-position code:

- D for Dictionary
- E for OLSD Screens (OSD)
- G PACBENCH/CS Screens (OSC)
- P for Batch Language Programs (BSD)

After two general programs (BVPACA10 and BVPACA20), common to all the chains, have been executed, the sub-chains are activated, according to the generation-print requests, in the following order:

- Screens
- Programs
- Dictionary

Each sub-chain performs an extraction (followed by a printing for GCP or GCO commands).

Once these sub-chains have been activated for the extraction of the entities to be analyzed, the BVPTUQ20 program performs the analysis according to the rules that it has been assigned and to the analysis parameters.

Results are printed by the BVPTUQ24, PBVTUQ25 and BVPTUQ30 programs.

The processing of the generated flow in the case of generation requests is identical to that of the GPRT procedure.

#### EXECUTION CONDITIONS

None. Files can remain open to on-line use.

#### **PQCA - User Input**

See the 'PQC' Reference manual.

#### **PQCA - Description of Steps**

INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS



## QUALITY ANALYSIS: PACQ

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV.&BASE.AR	Input	Development Database data
B7AN\$BASE	&INDUV.&BASE.AN	Input	Development Database index
B7AY\$BASE	&INDUV.&BASE.AY	Input	Development Database extension data
B7AJ\$BASE	&INDUV.&BASE.AJ	Input	Development Database Journal file
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
BVP7LB	&INDSV..BVPLB	Input	Generation follow-up
PACQMF	&PQCE	Input	Quality rules
PAC7SC	&INDSV..BVPSC	Input	Batch generation language skeleton
PAC7SG	&INDSV..BVPSC	Input	OLSD generation skeleton
PAC7SN	&INDSV..BVPSN	Input	Client/server meta-entity skeleton
PAC7SS	&INDSV..BVPSS	Input	Map skeleton
PAC7ME	&&PQCAMB	Input	Entity input to be analyzed
PACQMC	&&PQCAMC	Input	Selection parameter input
PAC7IA		Report	PACQ execution output
PAC7ID		Report	Documentation
PACQIB		Report	Selection parameter check
PACQIE		Report	Results by entity type
PACQIF		Report	Results by entity
PACQIG		Report	List of identifiers exceeding the identifiers limits
PAC7GB		Output	DBD generated Cobol
PAC7GE		Output	OLSD generated Cobol
PAC7GG		Output	Pacbench C/S generated Cobol
PAC7GP		Output	Batch language generated Cobol
PAC7GV		Output	PDM generated Cobol

Code	Physical name	Type	Label
PAC7GB	&INDUV..&BASE..GN&USER	Output	Concatenation of generated flows Other files mentioned in the procedure are temporary files used in the programs flows.
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### DELETION OF GENERATED FLOWS: IEFBR14

Code	Physical name	Type	Label
	DSN=&INDUV..&BASE..GN&USER	Input	Generated flows

#### PQCA - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*      - PACBENCH QUALITY CONTROL -
/*
/* -----
/*
/* THE PQCA PROCEDURE CARRIES OUT AN ANALYSIS OF THE
/* QUALITY OF THE APPLICATIONS, ACCORDING TO EITHER
/* STANDARD RULES OR USER-DEFINED RULES.
/*
/* -----
/*
//BVPPQCA PROC BASE=$BASE,          CODE OF VAPAC DATABASE
//  INDSV='$INDSV',                 INDEX OF SYSTEM VSAM FILES
//  INDSN='$INDSN',                 INDEX OF SYSTEM NON-VSAM FILES
//  INDUV='$INDUV',                 INDEX OF USER VSAM FILES
//  PQCE=,                           QUALITY CONTROL FILE DSN
//*: VSAMCAT='$VCAT',                USER VSAM CATALOG
//*: SYSCAT='$SCAT',                 VA PAC SYSTEM VSAM CATALOG
//  STEPLIB='$HLQ..SBVPMBR8',        LIBRARY OF LOAD-MODULES
//  SORTLIB='$BIBT',                 SORT LIBRARY
//  COPIES=1,                         NUMBER OF REPORT COPIES
//  USER=,                            USER CODE
//  OUT=$OUT,                          UTILITIES AND ERRORS OUTPUT CLASS
//  OUTL=$OUT,                          OUTPUT CLASS OF REPORTS
//  VOL='SER=$VOLUN',                 VOLUME OF GENERATION FILE
//  UNIT=$UNITUN,                     UNIT OF GENERATION FILE
//  UWK=$UWK,                          WORK UNIT
//  SPAEX='(TRK,(50,10),RLSE)',        SPACE OF PRINTING FILES
//  SPAWK='(TRK,(50,10))',            WORK FILE SPACE

```

```

//      SPAMB='(TRK,(5,1),RLSE)', REQUEST FILE SPACE
//      SPAGN='(TRK,(100,1),RLSE)', GENERATION FILE SPACE
//      PSBLIB='$PSBLIB', LIBRARY OF PSB'S
//      DBDLIB='$DBDLIB', LIBRARY OF DBD'S
//      RESLIB='$RESLIB', IMS RESLIB
//      PROCLIB='$PRCLIB', IMS PROCLIB
//      SPIE=0,TEST=0,NBA=,IN=,OUT1=,PRLD=,CKPTID=,DIRCA=000,
//      PARDLI=1,STIMER=,CPUTIME=,OBA=,AGN=,IMSID=$IMSID,OPT=N
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&PQCAMB,DISP=(,PASS),UNIT=&UWK,
// SPACE=(TRK,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//CARTE DD DDNAME=SYSIN
//*-----
//INPUT2 EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//PAC7MB DD DSN=&&PQCAMC,DISP=(,PASS),UNIT=&UWK,
// SPACE=&SPAMB,
// DCB=BLKSIZE=3440
//CARTE DD DDNAME=SYSIN2
//*-----
//VERIFY EXEC PGM=IDCAM5
//*:STEPDAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFLB),DISP=SHR
//*-----
//PACQ EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(BMP,BVPACQ,PACQ$SUG,&IN,&OUT1,
// &OPT&SPIE&TEST&DIRCA,&PRLD,&STIMER,&CKPTID,
// &PARDLI,&CPUTIME,&NBA,&OBA,
// &IMSID,&AGN)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR

```

```

//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7IA DD SYSOUT=&OUTL
//PAC7ID DD SYSOUT=&OUTL,COPIES=&COPIES
//PAC7IH DD SYSOUT=&OUTL
//PACQIB DD SYSOUT=&OUTL,DCB=BLKSIZE=133
//PACQIE DD SYSOUT=&OUTL,DCB=BLKSIZE=133
//PACQIF DD SYSOUT=&OUTL,DCB=BLKSIZE=133
//PACQIG DD SYSOUT=&OUTL,DCB=BLKSIZE=133
//PACQMC DD DSN=&PQCAMC,DISP=(OLD,DELETE,DELETE)
//PACQMF DD DSN=&PQCE,DISP=SHR
//PACQMJ DD UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=108,BLKSIZE=6588)
//PACQMK DD UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=196,BLKSIZE=6468)
//PACQMM DD UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=053,BLKSIZE=2120)
//PACQMN DD UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=053,BLKSIZE=2120)
//PACQMO DD UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=152,BLKSIZE=6080)
//PACQMZ DD UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=137,BLKSIZE=5480)
//BVP7LB DD DSN=&INDSV..BVPLB,DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7EE DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EP DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7EV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7JG DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3440
//PAC7KD DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KE DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820

```

```

//PAC7KF DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KG DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KP DD UNIT=&UWK,SPACE=&SPAEX,DCB=BLKSIZE=27820
//PAC7KS DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KU DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7KV DD UNIT=&UWK,SPACE=(TRK,(20,5),RLSE),DCB=BLKSIZE=27820
//PAC7ME DD DSN=&&PQCAMB,DISP=(OLD,DELETE,DELETE)
//PAC7MG DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=3060
//PAC7MV DD UNIT=&UWK,SPACE=(TRK,(2,1),RLSE),DCB=BLKSIZE=6800
//PAC7GB DD DSN=&INDUV..&BASE..GN&USER,
// DCB=BLKSIZE=12560,
// UNIT=&UNIT,
// VOL=&VOL,
// SPACE=&SPAGN,DISP=(,PASS,DELETE)
//PAC7GE DD DSN=*.PAC7GB,
// UNIT=&UNIT,
// VOL=REF=*.PAC7GB,
// DISP=(MOD,,DELETE)
//PAC7GG DD DSN=*.PAC7GB,
// UNIT=&UNIT,
// VOL=REF=*.PAC7GB,
// DISP=(MOD,,DELETE)
//PAC7GP DD DSN=*.PAC7GB,
// UNIT=&UNIT,
// VOL=REF=*.PAC7GB,
// DISP=(MOD,,DELETE)
//PAC7GV DD DSN=*.PAC7GB,
// UNIT=&UNIT,
// VOL=REF=*.PAC7GB,
// DISP=(MOD,,DELETE)
//PAC7SC DD DSN=&INDSV..BVVPS,DISP=SHR
//PAC7SG DD DSN=&INDSV..BVVPSG,DISP=SHR
//PAC7SS DD DSN=&INDSV..BVVPS,DISP=SHR
//PAC7SN DD DSN=&INDSV..BVVPSN,DISP=SHR
//PAC7W1 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W2 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W3 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27820
//PAC7W4 DD UNIT=&UWK,SPACE=&SPAWK,DCB=BLKSIZE=27180
//*-----
//DEL EXEC PGM=IEFBR14
//PAC7GN DD DSN=*.PACQ.PAC7GB,
// UNIT=&UNIT,
// VOL=REF=*.PACQ.PAC7GB,
// DISP=(OLD,DELETE,DELETE)

```

## Extraction of Quality Rules

### PQCE - Introduction

The PQCE procedure performs the extraction of quality rules created by the user in his/her database via the user entity supplied with the Customization option of the Pacbench Quality Control Facility.

It extracts the user entity occurrences that make up the customized quality rule dictionary, checks the information, and builds a file with the 'compiled' quality rules required by the Analysis of application quality (PQCA).

For further details, see the Pacbench Quality Control Reference Manual.

### EXECUTION CONDITIONS

None. The files can remain available for on-line use.

### **PQCE - Input / Processing / Results**

The user input of the PQCE procedure is similar to that of the EXUE extractor (PACX procedure).

One '\*' line per library to be consulted for extraction:

Position	Length	Value	Meaning
2	1	*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Library code
22	4	nnnn	Session number (Blank=current session)
26	1	T	Session status if Test session
28	1		not used
29	4	EXUE	Extractor code

For further details, see the chapter 'Manager's Utilities' (PACX: Extractions) in this manual.

One command line:

Pos.	Len.	Value	Meaning
2	4	W1EX	Line code
6	1	Y	UE extraction identifier
7	1	U C	Library selection code: Selected library Selected library + higher level lib.
8	2	5Q	Type code of user entity dedicated to Quality Control

### RESULT

The output of the PQCE procedure is a file containing the 'compiled' customized quality rules, which can be processed by the PQCA procedure.

PRINTED REPORT

This procedure prints:

- An occurrence-extraction report.
- A check report on the validity and usage of quality indicators.
- Descriptive reports on quality rules:
  - List of quality factors and criteria,
  - Description of each quality indicator,
  - Quality Control Dictionary.

**PQCE - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

INITIALIZATION OF THE KSDS WORKING FILE: IDCAMS

EXTRACTION: PACX

This step extracts transactions according to user input.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database Index file
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database Data file
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database Extension Data
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GU	&INDSV..BVPGU	Input	Administration Database Users
PAC7PJ	DUMMY	Input	Archived transactions
PAC7MB	&&PQCEMB	Input	User input
PAC7BM	&&PACXBM	Input/Output	User input

Code	Physical name	Type	Label
PAC7MM	&&PACXMM	Input/Output	EXPU work file
PAC7MJ	DUMMY	Input/Output	EXPJ work file
PAC7TE	DUMMY	Input/Output	RMEN work file
PAC7RE	DUMMY	Input/Output	RMEN work file
PAC7RM	DUMMY	Input/Output	RMEN work file
PAC7WD	&&PACXWD	Input/Output	Extracted transactions
SYSEXT	&INDUV..SYSEXT.&USER	Input/Output	Work file (indexed)
PAC7MV	DUMMY	Output	Extracted transactions for UPDT
PAC7MR	DUMMY	Output	Extracted transactions for REOR (EXPU)
PAC7GY	DUMMY	Output	Extracted transactions for UPDP
PAC7TD	DUMMY	Output	Extracted transactions for CPSN
PAC7UE	&&MB	Output	Extracted transactions for EXUE
PAC7IA		Report	General printing of program chains
PAC7DD		Report	Printing of abends on input transactions
PAC7EE		Report	Extraction reports
PAC7EP		Report	Extraction reports
PAC7EQ		Report	Extraction reports
PAC7EZ		Report	Extraction reports
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0: No error
- 4: Error on user input (detailed in PAC7EE) or on the extractions for EXTR/EXUE (detailed in PAC7EZ)
- 8: Error on '\*' line (detailed in PAC7DD) or in EXLI (Database not available)

#### COMPILATION OF QUALITY RULES: PTUQ10



This step creates the customized quality rule file that will be used by the PQCA analysis procedure.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PACQMI	&PQCF	Output	'Compiled' quality rules (length=80)
PAC7MB	&&PQCEMB	Input	User input
PACQMC	&&MB	Input	User input instances
PACQML	&&PACQML	Output	Preparation for printing
PACQIC		Report	Rule-validity report
PAC7DD		Report	Batch procedure authorization option
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### PRINTING OF QUALITY RULES: PTUQ15

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PACQML	&&PACQML	Input	Preparation for printing
PACQII		Report	List of quality factors and criteria and description by indicator
PACQIJ		Report	Dictionary of quality rules
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### DELETION OF THE KSDS WORKING FILE: IDCAMS

## PQCE - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*      - PACBENCH QUALITY CONTROL EXTRACTION -
/*
/* -----
/*
/* FORMAT OF TRANSACTIONS AT INPUT :
/* .. A USER AND LIBRARY LINE
/* .. A COMMAND LINE PER ENTITY TO BE EXTRACTED
/* COL 2-6  : "WIEX$"
/* COL 7    : SELECTION CODE OF THE LIBRARY
/*           "U"(LIBRARY ONLY)
/*           "C"(LIBRARY AND HIGHER LEVEL LIBRAIRIES)
/* COL 8-9  : TYPE CODE OF THE USER ENTITY (2 CHAR.)
/*
/* -----
/*
//BVPPQCE PROC BASE=$BASE,          CODE OF VAPAC DATABASE
//  INDSV='$INDSV',                 INDEX OF SYSTEM VSAM FILES
//  INDSN='$INDSN',                 INDEX OF SYSTEM NON-VSAM FILES
//  INDUV='$INDUV',                 INDEX OF USER VSAM FILES
//*:  VSAMCAT='$VCAT',              USER VSAM CATALOG
//*:  SYSCAT='$SCAT',              SYSTEM VSAM CATALOG
//  STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//  OUT=$OUT,                       OUTPUT CLASS
//  SPAMB='(TRK,(15,5))',          SPACE OF EXTRACTED TRANSACTIONS
//  UWK=$UWK,                        WORK UNIT
//  USER=,                          USER CODE
//  UNITS=$UNITUN,                  UNIT
//  VOLS='SER=$VOLUN',              QUALITY CONTROL FILE VOLUME
//  SPAPQC='(TRK,(60,15),RLSE)',   SPACE OF QUALITY CONTROL FILE
//  SORTLIB='$BIBT',               SORT LIBRARY
//  CYL=3,                          SORTWORK SPACE
//  OUTL=$OUT,                      OUTPUT CLASS OF REPORTS
//  PSBLIB='$PSBLIB',              LIBRARY OF PSB'S
//  DBDLIB='$DBDLIB',              LIBRARY OF DBD'S
//  RESLIB='$RESLIB',              IMS RESLIB
//  PROCLIB='$PRCLIB',             IMS PROCLIB
//  BUF=40, SPIE=0, TEST=0, EXCPVR=0, RST=0, PRLD=, SRCH=0,
//  CKPTID=, MON=N, LOGA=0, FMTO=T, DBRC=$DBRC, IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//CARTE  DD DDNAME=SYSIN
//PAC7MB DD DSN=&&PQCEMB,DISP=(,PASS),
//        UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//-----
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER'
//-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR

```

```

//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLPQCE),DISP=SHR
//PACROU DD DSN=&&DLPQCE,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLPQCE,DISP=(OLD,DELETE)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER'
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----
//DEFINE EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//*-----
//MAXKEY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF DD DSN=&&SYSEXT,DISP=(NEW,PASS),
//          SPACE=(CYL,(3,3)),
//          LRECL=254,RECOG=KS,KEYOFF=0,KEYLEN=43
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//*-----
//PACX EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPACX,PACX$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR

```

```

//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE DD DSN=&INDSV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDSV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDSV..&BASE.AY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7PJ   DD DUMMY
//PAC7IA   DD SYSOUT=&OUTL
//PAC7DD   DD SYSOUT=&OUTL
//PAC7EE   DD SYSOUT=&OUTL
//PAC7EP   DD SYSOUT=&OUTL
//PAC7EQ   DD SYSOUT=&OUTL
//PAC7EZ   DD SYSOUT=&OUTL
//SYSEXT  DD DSN=&&SYSEXT,DISP=(OLD,DELETE,DELETE)
//PAC7MB   DD DSN=&&PQCEMB,DISP=(OLD,PASS)
//PAC7BM   DD DSN=&&PACXBM,DISP=(,DELETE),UNIT=&UWK,
//          DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7MM   DD DSN=&&PACXMM,DISP=(,DELETE),UNIT=&UWK,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=55,BLKSIZE=6215)
//PAC7MJ   DD DUMMY
//PAC7TE   DD DUMMY
//PAC7RE   DD DUMMY
//PAC7RM   DD DUMMY
//PAC7WD   DD DSN=&&PACXWD,DISP=(,DELETE),UNIT=&UWK,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7MV   DD DUMMY
//PAC7MR   DD DUMMY
//PAC7MX   DD DUMMY
//PAC7TD   DD DUMMY
//PAC7GY   DD DUMMY

```

```

//PAC7UE DD DSN=&&MB,DISP=(,PASS),UNIT=&UWK,
//          SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=230,BLKSIZE=6440)
//*-----
//PTUQ10 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUQ10,PTUQ10$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=(00,NE,PACX)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),,CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7MB DD DSN=&&PQCEMB,DISP=(OLD,DELETE)
//PACQMC DD DSN=&&MB,DISP=(OLD,DELETE)
//PACQMI DD DSN=&INDUV..PQCE.&USER,DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,VOL=&VOLS,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
//          SPACE=&SPAPQC
//PACQML DD DSN=&&PACQML,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(15,5),RLSE),
//          DCB=(RECFM=FB,LRECL=85,BLKSIZE=8500)
//PAC7DD DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PACQIC DD SYSOUT=&OUT,DCB=BLKSIZE=133
//*-----
//PTUQ15 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUQ15,PTUQ15$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,

```

```

//      &IRLM),
//      COND=( (00,NE,PACX), (00,NE,PTUQ10) )
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSOUX  DD SYSOUT=&OUT
//DDSNAP  DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON  DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),CONTIG)
//BVP7AE  DD DSN=&INDSV..BVPAE,DISP=SHR
//PACQML  DD DSN=&&PACQML,DISP=(OLD,DELETE)
//PACQII  DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PACQIJ  DD SYSOUT=&OUT,DCB=BLKSIZE=133
//*-----
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER',COND=EVEN
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//PACRIN  DD DSN=&INDSN..BVPSY(DLSYSEXT),DISP=SHR
//PACROU DD DSN=&&DLSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//      UNIT=&UWK,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN   DD DSN=&&DLSYSEXT,DISP=(OLD,DELETE)

```

---

## Chapter 6. Versioning Facilities

---

### Standard Bridge (PCM)

#### Introduction

The SCM (SUPPORT CONFIGURATION MANAGEMENT tools interface) module is optional. Its use depends upon the . corresponding purchase agreement. The purpose of the module is twofold:

#### 1. PRODUCTION TURNOVER MANAGEMENT

This function is used to:

- Manage the generation environments, by specifying the 'production environments' which are used to manage the database freeze.
- Follow up entities generated from a database and put into production.
- Give the user information related to these entities such as their library code, the session number of the last generation, and of the last database freeze.
- Automatically freeze the database when generating into a production environment.
- Provide project follow-up to development teams in relation to generated entities.

#### 2. RECOGNITION OF A CONFIGURATION MANAGEMENT UTILITY

With the use of two complementary procedures, the module ensures the consistency between the production turnover data which is stored in the development Database, and the programs generated by a configuration management utility. But the user needs to ship a file that is to be extracted from the product so as to compare it with another file extracted from the development Database.

#### Database Automatic Freeze

##### **HIPM - Introduction**

The purpose of the HIPM procedure is to generate transactions related to the production turnover of entities and if necessary, the transactions related to the freezing of the development Database.

##### EXECUTION CONDITIONS

None

## ABNORMAL EXECUTION

The procedure can be restarted once the problem has been solved.

### **HIPM - Input / Processing / Results**

A '\*' line with user code and password.

User input specific to the optional procedure which is used for database freeze request.

The structure of the line is as follows:

Position	Length	Value	Meaning
2	2		Line code
		'X1'	if the entities have been put into production
		'X4'	If no entity has been put into production
4	4	'HIST'	Freeze request
8	50		Freeze comments
58	4	ssss	Forcing of session number to be frozen: this number must be included between the current session number +1 and the current session +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 to be frozen.

## PRINTED REPORTS

This procedure prints:

- a report,
- a list of the entities used in production, and if the database has been frozen.

## RESULTS

The result of the procedure execution is a sequential file containing the production environment transactions and the optional freeze transactions.



This file must be put in input of the UPDP procedure in order to update the development Database.

**HIPM - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

DELETION OF PRODUCTION TURNOVER TRANSACTIONS: IDCAMS

This step deletes the production turnover transaction file, if it already existed.

This file will be allocated in the next step.

ALLOCATING PRODUCTION TURNOVER TRANSACTIONS: IEFBR14

The purpose of this step is to allocate the file that will contain the production turnover transactions:

ALLOCMY : DSN=&INDUN.&BASE.MY

GENERATION OF PRODUCTION TURNOVER TRANSACTIONS: PCM300

This step is used to explore the development database and generate production turnover and database freeze transactions.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database random data
PAC7MB	&&HIPMMB	Input	Users data .
PAC7TR	&&HIPMTR	Output	Work file
PAC7SR	&&HIPMSR	Output	Work file
PAC7IG		Report	Production turnover output report
PAC7GY	&INDUN..&BASE.MY	Output	Production turnover transactions

Code	Physical name	Type	Label
PAC7DD		Report	Batch procedures authorization option

## HIPM - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/* AUTOMATIC SESSION FREEZE
/* -----
/*
/* INPUT      : USER IDENTIFICATION
/* COL 2      : "*"
/* COL 3      : USER CODE
/* COL 11     : PASSWORD
/* -----
//BVPHIPM  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',                INDEX OF SYSTEM NON VSAM FILES
//          INDUV='$INDUV',                INDEX OF USER VSAM FILES
//          INDUN='$INDUN',                INDEX OF USER NON VSAM FILES
//*:       VSAMCAT=$VCAT',                USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//          OUT=$OUT,                       UTILITIES AND ERRORS OUTPUT CLASS
//          UWK=$UWK,                         WORK UNIT
//          SPATR='(TRK,(1,1),RLSE)',        REQUEST SPACE TR FILE
//          SPASR='(TRK,(1,1),RLSE)',        REQUEST SPACE SR FILE
//          SPAMY='(TRK,(1,1),RLSE)',        REQUEST SPACE MY FILE
//          PSBLIB='$PSBLIB',                LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',                LIBRARY OF DBD'S
//          RESLIB='$RESLIB',                IMS RESLIB
//          PROCLIB='$PRCLIB',                IMS PROCLIB
//          BUF=40, SPIE=0, TEST=0, EXCPVR=0, RST=0, PRLD=, SRCH=0,
//          CKPTID=, MON=N, LOGA=0, FMTO=T, DBRC=$DBRC, IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB, DISP=SHR
//          DD DSN=$BCOB, DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&HIPMMB, DISP=(,PASS),
//          UNIT=&UWK, SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB, LRECL=80, BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT, DISP=SHR
//*:          DD DSN=&SYSTCAT, DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN  DD DSN=&INDUV..&BASE.AN, DISP=SHR

```

```

//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGY DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//DELMY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.MY),DISP=SHR
//ALLOCMY EXEC PGM=IEFBR14
//*-----
//SYSPRINT DD SYSOUT=&OUT
//ALLOCMY DD DSN=&INDUN..&BASE.MY,
// DISP=(,CATLG),
// UNIT=$UNITUN,
// VOL=SER=$VOLUN,
// DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200),
// SPACE=&SPAMY
//*-----
//PCM300 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPCM300,PCM300$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR

```

```

//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7IG DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MB DD DSN=&&HIPMMB,DISP=(OLD,DELETE)
//PAC7DD DD SYSOUT=&OUT
//PAC7TR DD DSN=&&HIPMTR,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPATR,
// DCB=(RECFM=FB,LRECL=1370,BLKSIZE=27400)
//PAC7SR DD DSN=&&HIPMSR,DISP=(,PASS),
// UNIT=&UWK,SPACE=&SPASR,
// DCB=(RECFM=FB,LRECL=1370,BLKSIZE=27400)
//PAC7GY DD DSN=&INDUN..&BASE.MY,DISP=SHR

```

## Generation Simulation

### SIPM - Introduction

The SIPM procedure is used to simulate the production turnover of entities which is usually performed by GPRT.

There are two possibilities:

- Production turnover of entities:

The information related to the entities and the environment is entered by the user.

- Transfer from one environment to another one:

The information related to the entity is provided by the source environment.

### EXECUTION CONDITIONS

None.

The files can remain open.

### ABNORMAL EXECUTION

Whatever the reason for an abnormal ending, the procedure can be restarted as it is, once the problem has been solved.

### SIPM - Input / Processing / Results

A '\*' line with user code and password including information related to the procedure

Position	Length	Value	Meaning
2	1	'*'	Line code

Position	Length	Value	Meaning
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code
22	4	ssss	session number (blank if current session)
26	1		Session status (' ' or 'T')
59	8	ccyymmdd	Generation date , if the session is not the current session (input field for a frozen session of blank or T type, no input field for a current session)

An environment identification line 'EG' (required):

Position	Length	Value	Meaning
2	2	'EG'	Line code
4	3	ttt	Type of entities processed
7	30		Target environment
37	10		Target application

Source environment line 'ES'(if transfer):

Position	Length	Value	Meaning
2	2	'ES'	Line code
7	30		Source environment
37	10		Source application

Entity identification line 'EU' for each entity generation to be simulated

Position	Length	Value	Meaning
2	2	'EU'	Line code
4	6	cccccc	Entity code
10	8	eeeeeeee	External name of the entity in target environment (if different from the database code)
18	8	nnnnnnnn	External name of the entity in source environment (if transfer with RENAME)

REPORT OUTPUT

This procedure prints a report

RESULTS

Once the procedure has been executed, a sequential file is produced. This file contains the production turnover transactions.

It must be used as input to the UPDP procedure in order to update the development Database.

**SIPM - Description of Steps**  
INPUT RECOGNITION: PTU001

WORK FILE RRDS INITIALIZATION: PCMIWY

Code	Physical name	Type	Label
PAC7WY	&INDUV..&BASE.WY	Output	Work file RRDS

DELETION OF TRANSACTIONS FILE FOR UPDP: IDCAMS

Code	Physical name	Type	Label
	&INDUN..&BASE.MY	Output	Update transactions file

ALLOCATION OF TRANSACTIONS FILE: IEFBR14

Code	Physical name	Type	Label
ALLOCMY	&INDUN..&BASE.MY	Output	Update transactions file

VSAM FILES CHECK: IDCAMS

GENERATION OF SIMULATION TRANSACTIONS: PCM320

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AY\$BASE	&INDUV..&BASE.AY	Input	Administration Database random data

Code	Physical name	Type	Label
PAC7MB	&&SIPMMB	Input	User transactions
PAC7MT	&&SIPMMT	Output	File to be used by a transfer utility
PAC7IE		Report	Simulation output report
PAC7GY	&INDUV..&BASE.WY	Output	Transactions for UPDP (simulation output)
PAC7DD		Report	Batch procedures authorization option

### CREATION OF TRANSACTIONS FOR UPDP PROCEDURE: IDCAMS

Code	Physical name	Type	Label
IN1	&INDUV..&BASE.WY	Input	RRDS transactions file
OU1	&INDUN..&BASE.MY	Output	UPDP transactions file

### **SIPM - Execution JCL**

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/* SIMULATION
/* -----
/*
/* INPUT      : USER IDENTIFICATION
/* COL 2      : "*"
/* COL 3      : USER CODE
/* COL 11     : PASSWORD
/* COL 19     : LIBRARY
/* COL 22     : SESSION
/* COL 25     : SESSION STATE
/* -----
//BVPSIPM PROC BASE=$BASE,          CODE OF DEVPT DATABASE
//      INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//      INDSN='$INDSN',             INDEX OF SYSTEM ON VSAM FILES
//      INDUV='$INDUV',             INDEX OF USER VSAM FILES
//      INDUN='$INDUN',             INDEX OF USER NON VSAM FILES
/**:      VSAMCAT='$SVCAT',          SYSTEM VSAM CATALOG
/**:      SYSCAT='$SCAT',           SYSTEM VSAM CATALOG
//      STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//      VOLUN='$VOLUN',             VOLUME OF UPDP TRANSACTIONS
//      OUT=$OUT,                   OUTPUT CLASS
//      UWK=$UWK,                   WORK UNIT
//      SPAMT='(TRK,(1,1),RLSE)',   SPACE OF MT FILE
//      SPAMY='(TRK,(10,1),RLSE)',  SPACE OF MY FILE
//      PSBLIB='$PSBLIB',           LIBRARY OF PSB'S

```

```

//          DBDLIB='$DBDLIB',                LIBRARY OF DBD'S
//          RESLIB='$RESLIB',                IMS RESLIB
//          PROCLIB='$PRCLIB',              IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&SIPMMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//ALLOCWY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DF&BASE.WY),DISP=SHR
//PCMIWY EXEC PGM=BVPCMIWY
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7WY DD DSN=&INDUV..&BASE.WY,DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//DELMY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.MY),DISP=SHR
//ALLOCMY EXEC PGM=IEFBR14
//*-----
//SYSPRINT DD SYSOUT=&OUT
//ALLOCMY DD DSN=&INDUN..&BASE.MY,
//          DISP=(,CATLG),
//          UNIT=&UWK,
//          VOL=SER=&VOLUN,
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200),
//          SPACE=&SPAMY
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR

```



```

//          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//*-----
//PC320 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPC320,PC320$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDR DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7MB DD DSN=&&SIPMMB,DISP=(OLD,DELETE,DELETE)
//PAC7DD DD SYSOUT=&OUT
//PAC7IE DD SYSOUT=&OUT
//PAC7MT DD DSN=&&SIPMT,DISP=(,PASS),
//      UNIT=&UWK,SPACE=&SPAMT,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
//PAC7GY DD DSN=&INDUV..&BASE.WY,DISP=SHR
//MVUPDP EXEC PGM=IDCAMS
//*-----
//IN1 DD DSN=&INDUV..&BASE.WY,DISP=SHR
//OU1 DD DSN=&INDUN..&BASE.MY,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=$INDSN..BVPSY(REWYMY),DISP=SHR

```

## Extraction of the Development Database Data

### EXPM - Introduction

The EXPM procedure extracts, from the development Database, the entities whose generation status is to be compared to the configuration management utility.

The extracted file will be compared to the file extracted from the utility.

The extraction can be limited to sessions, databases, environments and applications.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

The procedure can be restarted as it is once the problem has been solved.

**EXPM - Input / Processing / Results**

A '\*' line with user code and password.

Position	Length	Value	Meaning
2	1	'*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code
		***	if extraction from all Libraries
22	4	ssss	Number of session in use (blank in current session)
		****	if extraction from all sessions
26	1	*	Status of the session ( ' ' or 'T') if extraction from all sessions

One or more 'S' line(s) to select the environments /application

The line is structure is as follows:

Position	Length	Value	Meaning
2	1	'S'	Line code
3	30		Selected environment
33	10		Selected application

PRINTED REPORT

This procedure prints a report.

## RESULTS

Once the procedure has been executed, a sequential file extracted from the development Database is produced, it must be used as input to the CPPM procedure.

## IMPORTANT INFORMATION

Before executing the procedure, you must allocate a sequential file resulting from the extraction.

This file is due to receive the extracted elements.

Characteristics:

Length: 900 characters

It will have to be entered in the execution procedure JCL by assigning its DSNNAME to the EXTRFILE parameter.

### **EXPM - Description of Steps** INPUT RECOGNITION: PTU001

### VSAM FILES CHECK: IDCAMS

### EXTRACTION OF THE DEVELOPMENT DATABASE: PCM200

This step explores the development Database and extracts the elements in accordance with the extraction request.

Code	Physical name	Type	Label
PAC7MV	&&EXPMMB	Input	User transactions
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database random data
PAC7ET		Report	Extraction output report
PAC7MS	&&EXPMMS01	Output	Extracted elements file

Code	Physical name	Type	Label
PAC7BM	&&EXPMBM	Output	Working file
PAC7DD		Report	Batch procedures authorization option
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### DELETION OF DUPLICATE EXTRACTED ELEMENTS: PCM202

This step is used to suppress the elements that would be assigned by error to several extracted applications.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7ME	&&EXPMMS01	Input	Input extracted elements file
PAC7MS	&&EXPMMS02	Output	Output extracted elements file
PAC7EQ		Report	List of elements extracted twice
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### EXTRACTED ELEMENTS SORTING FILE: PCM205

This step sorts the extracted elements file in accordance with the criterion required for the comparison with the file provided by the configuration management utility.

Code	Physical name	Type	Label
PAC7ME	&&EXPMMS02	Input	Input extracted elements file
PAC7MS	&EXTRFILE	Output	Output extracted elements file
SORTWK02		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

## EXPM - Execution JCL

```

//* -----
//*      VISUALAGE PACBASE
//*
//* -----
//* EXTRACTION
//* -----
//*
//* INPUT      : USER IDENTIFICATION
//* COL 2     : "*"
//* COL 3     : USER CODE
//* COL 11    : PASSWORD
//* COL 19    : LIBRARY
//* COL 22    : SESSION
//* COL 26    : SESSION STATE
//* -----
//BVPEXPM  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          INDSV='$INDSV',                INDEX OF VSAM SYSTEM FILES
//          INDSN='$INDSN',                INDEX OF NON VSAM SYSTEM FILES
//          INDUV='$INDUV',                INDEX OF VSAM USER FILES
//*:       VSAMCAT='$VCAT',                USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',              SORT LIBRARY
//          OUT=$OUT,                      OUTPUT CLASS
//          UWK=$UWK,                      WORK UNIT
//          EXTRFILE=' ',                  VA PAC EXTRACTED FILE
//                                           (LENGTH = 900)
//          CYL=(1,1)',                   TEMPORARY SPACE
//          SPAMB=(TRK,(1,1),RLSE)',      SPACE OF MB FILE
//          SPAMS=(TRK,(1,1),RLSE)',      SPACE OF MS FILE
//          PSBLIB='$PSBLIB',             LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//          RESLIB='$RESLIB',            IMS RESLIB
//          PROCLIB='$PRCLIB',           IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&EXPMMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY  DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR

```

```

//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//*-----
//PCM200 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPCM200,PCM200$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVP7GU,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7ET   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
//PAC7MV   DD DSN=&&EXPMMB,DISP=(OLD,PASS)
//PAC7MS   DD DSN=&&EXPMS01,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=&SPAMS,
//          DCB=(RECFM=FB,LRECL=900,BLKSIZE=27000)
//PAC7BM   DD DSN=&&EXPMBM,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=&SPAMB,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
//*-----
//PCM202 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPCM202,PCM202$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR

```

```

//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7EQ   DD SYSOUT=&OUT
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7ME   DD DSN=&&EXPMS01,DISP=SHR
//PAC7MS   DD DSN=&&EXPMS02,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=&SPAMS,
//          DCB=(RECFM=FB,LRECL=900,BLKSIZE=27000)
//*-----
//PCM205 EXEC PGM=BVPCM205
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7ME   DD DSN=&&EXPMS02,DISP=SHR
//PAC7MS   DD DSN=&EXTRFILE,DISP=SHR

```

## Comparison with Extracted Files

### CPPM - Introduction

The CPPM procedure is used to compare a file extracted from the development Database via the EXPM procedure, with a similar file extracted by the user from the configuration management utility.

The purpose is to generate a transactions file to update the development Database via the UPDP procedure.

The transactions are due to set the development Database to the level of the configuration management utility, in relation to the production entities.

### EXECUTION CONDITIONS

The EXPM procedure must be executed at first so as to obtain a file extracted from the development Database.

Moreover, a file extracted from the configuration management utility must have been defined, with the same status as the file's extracted from the development Database

### ABNORMAL EXECUTION

Whatever the reason for an abnormal ending, the procedure can be restarted as it is, once the problem has been solved.

### **CPPM - Input / Processing / Results**

A '\*' line with user code and password.

### PRINTED REPORT

This procedure prints:

- A report,
- A list of entities which will be modified in the development Database after executing the UPDP procedure.

### RESULTS

Once the procedure executed, a sequential file is produced. It contains the development Database update transactions to be used as input to the UPDP procedure.

### **CPPM - User File**

To be able to align the development Database with the Configuration management utility used on-site, it is necessary to create a file which contains the data extracted from the utility so as to compare it with the file extracted from the development Database via the EXPM procedure.

The length of the file should be 900 with the following structure:

Position	Length	Value	Meaning
1	35		Parameter 1
36	35		Parameter 2
71	35		Parameter 3



Position	Length	Value	Meaning
106	35		Parameter 4
141	35		Parameter 5
176	35		Parameter 6
211	35		Parameter 7
246	35		Parameter 8
281	35		Parameter 9
316	35		Parameter 10
351	35		Parameter 11
386	35		Parameter 12
421	35		Parameter 13
456	35		Parameter 14
491	35		Parameter 15
526	30		Environment code
556	10		Application code
566	1		Entity type
567	6		Entity
573	8		External code of entity
585	3		Library code
588	4		Session number
592	1		Session status
593	2		Call code (when it is a user entity)
595	10		Generation data (CCYYMMDD)
605	8		Generation time
613	8		User code
621	35		Group field
796	51		Optional fields which display information on the last action performed on the object:
796	20		Action label
816	10		Action date
826	8		Action hour
834	8		User code
842	5		Object version

Position	Length	Value	Meaning
847	54		VA Pac internal fields

The information of 'parameter 1' up to 'parameter 15' corresponds to the parameters defined in the environment screen. The sorting order is taken into account.

The 'Entity type' and the next information correspond to the values defined in the generated program, in the PACBASE-CONSTANTS (or CONSTANTES-PACBASE).

**CPPM - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

ALLOCATION OF RRDS WORKING FILE: IDCAMS

Code	Physical name	Type	Label
	&INDUV.&BASE.WY	Output	Working file

WORK FILE INITIALIZATION: PCMIWY

Code	Physical name	Type	Label
PAC7QJ	&INDUV.&BASE.WY	Output	RRDS work file

DELETE OF THE TRANSACTIONS FILE: IDCAMS

Code	Physical name	Type	Label
	&INDUN.&BASE.MY	Output	Update transactions file

ALLOCATION OF THE TRANSACTIONS FILE: IEFBR14

Code	Physical name	Type	Label
ALLOCMY	&INDUN.&BASE.MY	Output	Update transactions file

COMPARISON PROCESSING: PCM210

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7MB	&&CPPMMB	Input	User transactions
PAC7MP	&EXTRFILE	Input	File extracted from the development Database
PAC7MU	&TOOLFILE	Input	File extracted from configuration management utility
PAC7EQ		Report	Output report of check
PAC7ME	&&CPPMME	Output	File used to print comparison errors
PAC7MS	&&CPPMMS	Output	File used to print update transactions
PAC7DD		Report	Batch procedures authorization option
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

PRINT OF UPDATE TRANSACTIONS: PCM220

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7EQ		Report	Comparison output report
PAC7MS	&&CPPMMS	Input	File used to print the results of comparison
PAC7GY	&INDUV..&BASE.WY	Input	RRDS file containing transactions for UPDP
SORTWK01		Sort	
SORTWK02		Sort	

Code	Physical name	Type	Label
SORTWK03		Sort	

### CREATION OF TRANSACTIONS FOR UPDP PROCEDURE: IDCAMS

Code	Physical name	Type	Label
IN1	&INDUV.&BASE.WY	Output	RRDS transactions file
OU1	&INDUN.&BASE.MY	Output	UPDP transactions file

### CPPM - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/* COMPARISON
/* -----
/*
/* INPUT      : USER IDENTIFICATION
/* COL 2      : "*"
/* COL 3      : USER CODE
/* COL 11     : PASSWORD
/* -----
//BVPCPPM PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//      INDSV='$INDSV',                    INDEX OF SYSTEM VSAM FILES
//      INDSN='$INDSN',                    INDEX OF SYSTEM NON VSAM FILES
//      INDUV='$INDUV',                    INDEX OF USER VSAM FILES
//      INDUN='$INDUN',                    INDEX OF USER NON VSAM FILES
//*:      VSAMCAT='$VCAT',                  USER VSAM CATALOG
//*:      SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//      STEPLIB='$HLQ..SBVPMBR8',          LIBRARY OF LOAD-MODULES
//      SORTLIB='$BIBT',                    SORT LIBRARY
//      EXTRFILE=' ',                       VA PAC EXTRACTED FILE
//      TOOLFILE=' ',                       TOOL EXTRACTED FILE
//      OUT=$OUT,                           OUTPUT CLASS
//      UWK=$UWK,                           WORK UNIT DE TRAVAIL
//      SPAMY='(TRK,(1,1),RLSE)',          SPACE OF MY FILE
//      CYL='(1,1)',                        TEMPORARY SPACE
//      PSBLIB='$PSBLIB',                   LIBRARY OF PSB'S
//      DBDLIB='$DBDLIB',                   LIBRARY OF DBD'S
//      RESLIB='$RESLIB',                   IMS RESLIB
//      PROCLIB='$PRCLIB',                  IMS PROCLIB
//      BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//      CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
/*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//CARTE  DD DDNAME=SYSIN
//PAC7MB DD DSN=&CPPMMB,DISP=(,PASS),

```

```

//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//      DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//      DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//ALLOCWY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DF&BASE.WY),DISP=SHR
//PCMIWY EXEC PGM=BVPCMIWY
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PAC7WY DD DSN=&INDUV..&BASE.WY,DISP=SHR
//SYSUDUMP DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//DELMY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.MY),DISP=SHR
//ALLOCMY EXEC PGM=IEFBR14
//*-----
//SYSPRINT DD SYSOUT=&OUT
//ALLOCMY DD DSN=&INDUN..&BASE.MY,
//      DISP=(,CATLG),
//      UNIT=$UNITUN,
//      VOL=SER=$VOLUN,
//      DCB=(RECFM=FB,LRECL=310,BLKSIZE=6200),
//      SPACE=&SPAMY
//*-----
//PCM210 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPCM210,PCM210$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//      &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR

```

```

//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT     DD SYSOUT=&OUT
//SYSOUX     DD SYSOUT=&OUT
//DDSNAP     DD SYSOUT=&OUT
//PROCLIB    DD DSN=&PROCLIB,DISP=SHR
//IEFRDER    DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP   DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP   DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON     DD DUMMY
//DFSVSAMP   DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB    DD DSN=&SORTLIB,DISP=SHR
//SORTWK01   DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02   DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03   DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE     DD DSN=&INDSV..BVP AE,DISP=SHR
//BVP7GN     DD DSN=&INDSV..BVP GN,DISP=SHR
//BVP7GR     DD DSN=&INDSV..BVP GR,DISP=SHR
//BVP7GU     DD DSN=&INDSV..BVP GU,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7EQ     DD SYSOUT=&OUT,DCB=BLKSIZE=133
//PAC7MP     DD DSN=&EXTRFILE,DISP=SHR
//PAC7MU     DD DSN=&TOOLFILE,DISP=SHR
//PAC7ME     DD DSN=&&CPPMME,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=1000,BLKSIZE=30000)
//PAC7MS     DD DSN=&&CPPMMS,DISP=(NEW,PASS),
//          UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=400,BLKSIZE=32000)
//PAC7MB     DD DSN=&&CPPMMB,DISP=SHR
//PAC7DD     DD SYSOUT=&OUT
//*-----
//PCM220     EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPCM220,PCM220$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB    DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB   DD DSN=&RESLIB,DISP=SHR
//IMS        DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT     DD SYSOUT=&OUT
//SYSOUX     DD SYSOUT=&OUT
//DDSNAP     DD SYSOUT=&OUT
//PROCLIB    DD DSN=&PROCLIB,DISP=SHR
//IEFRDER    DD DUMMY,

```

```

//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7EQ   DD SYSOUT=&OUT
//PAC7MS   DD DSN=&CPPMMS,DISP=SHR
//PAC7GY   DD DSN=&INDUV..&BASE.WY,DISP=SHR
//MVUPDP   EXEC PGM=IDCAMS
//*------
//IN1 DD DSN=&INDUV..&BASE.WY,DISP=SHR
//OU1 DD DSN=&INDUN..&BASE.MY,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDSN..BVPSY(REWYMY),DISP=SHR

```

## Integrity Control of Events/Elements

### CHPM - Introduction

The CHPM procedure is used to check the consistency of the environments and elements present in the VA Pac Database. The procedure produces a report printout for erroneous events and elements. The purpose of this check is to highlight the inconsistencies in the development Database.

### EXECUTION CONDITION

None.

### ABNORMAL EXECUTION

The procedure can be restarted as it is, once the problem has been solved.

### CHPM - Input / Processing / Results

Batch procedures access authorization option: A '\*' line with user code and password.

### PRINTED REPORT

This procedure prints a report listing the consistency errors detected in the development Database, and related to the environments and elements.

**CHPM - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

ENVIRONMENTS/ELEMENTS CONSISTENCY CHECK: PCM400

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database random data
PAC7MB	&&CHPMMB	Input	User transactions
PAC7MS	&&CHPMMS	Output	File used to print a check report
PAC7MV	&&CHPMMV	Output	Working file
PAC7DD		Report	Batch procedures authorization option
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

CONSISTENCY CHECK REPORT PRINTOUT: PCM410

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7EQ		Report	Output validation report
PAC7MS	&&CHPMMS	Input	File used for the report printout
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data

**CHPM - Execution JCL**

```

//* -----
//*      VISUALAGE PACBASE
//*
//* -----

```



```

/* VALIDATION OF THE DEVELOPMENT DATABASE
/* -----
/*
/* INPUT      : USER IDENTIFICATION
/* COL 2     : "*"
/* COL 3     : USER CODE
/* COL 11    : PASSWORD
/* -----
//BVPCHPM  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          INDUV='$INDUV',                INDEX OF USER VSAM FILES
//          INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',                INDEX OF SYSTEM NON VSAM FILES
//*:        VSAMCAT='$VCAT',                USER VSAM CATALOG
//*:        SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//          OUT=$OUT,                       OUTPUT CLASS
//          UWK=$UWK,                       WORK UNIT
//          SPAMV='(TRK,(30,10),RLSE)',    SPACE OF MV FILE
//          SPAMS='(TRK,(30,10),RLSE)',    SPACE OF MS FILE
//          CYL='(1,1)',                    TEMPORARY SPACE
//          PSBLIB='$PSBLIB',              LIBRARY OF PSB'S
//          SORTLIB='$BIBT',              SORT LIBRARY
//          DBDLIB='$DBDLIB',             LIBRARY OF DBD'S
//          RESLIB='$RESLIB',             IMS RESLIB
//          PROCLIB='$PRCLIB',            IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&CHPMMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//*-----
//PCM400 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPCM400,PCM400$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR

```

```

//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVP7GU,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7MB   DD DSN=&&CHPMMB,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7IE   DD SYSOUT=&OUT
//PAC7MV   DD DSN=&&CHPMMV,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPAMV,
//          DCB=(RECFM=FB,LRECL=2660,BLKSIZE=26600)
//PAC7MS   DD DSN=&&CHPMMS,DISP=(,PASS),
//          UNIT=&UWK,SPACE=&SPAMS,
//          DCB=(RECFM=FB,LRECL=2660,BLKSIZE=26600)
//*-----
//PCM410   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPCM410,PCM410$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT

```

```

//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7EQ   DD SYSOUT=&OUT
//PAC7MS   DD DSN=&&CHPMMS,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR

```

## Update

### UPPM - Introduction

The processing starts from the first transaction of QJ not processed. It prepares for the communication area to be filled in with PAF-type transactions extracted from QJ before the call of PUF for the update. The call of the PUF function is made during the update of a PAF file, modification of the user code or of the Database code.

The PUF returned errors, if any, are saved in the QJ file.

In case of system error related to the Database, the Database transactions are ignored during the processing.

During the archiving, the valid transactions are deleted from the QJ file while the erroneous transactions or the transactions which have not been processed are recycled to reconstitute the new QJ file for the next processing.

### EXECUTION CONDITION

The QJ file must exist.

### UPPM - Input / Processing / Results

The UPPM procedure updates the VA Pac Databases from the QJ file which contains PAF Table formatted transactions written by the GPRT generation procedure during generation.

### OUTPUT REPORTS

None.

### UPPM - Description of Steps

UPDATE : PCMPUF

This step updates the User Entities of the PCM Meta-Entity in the VA Pac Database.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database random data
B7DC\$BASE	&DSMS..&BASE.DC	Input	DSMS Control data
B7D3\$BASE	&DSMS..&BASE.D3	Input	DSMS Control data index
PAC7QJ	&INDSV..BVPQJ(0)	Input	Standard bridge journal file

### UPPM - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*      - PCM UPDATE -
/*
/*
/* -----
//BVPUPPM  PROC BASE=$BASE,          CODE OF VAPAC DATABASE
//          INDSV='$INDSV',          INDEX OF VSAM SYSTEM FILES
//          INDSN='$INDSN',          INDEX OF NO VSAM SYSTEM FILES
//          INDUV='$INDUV',          INDEX OF VSAM USER FILES
//          DSMS='$DSMS',            INDEX OF DSMS FILES
//*:       VSAMCAT='$VCAT',          USER VSAM CATALOG
//*:       SYSCAT='$SCAT',           SYSTEM VSMA CATALOG
//          OUT='$OUT',              UTILITIES AND ERROR OUPUT CL
//          STEPLIB='$HLQ..SBVPMTR8', LIBRARY OF LOAD MODULES
//          PSBLIB='$PSBLIB',        LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',        LIBRARY OF DBD'S
//          RESLIB='$RESLIB',         IMS RESLIB
//          PROCLIB='$PRCLIB',        IMS PROCLIB
//          SPIE=0,TEST=0,NBA=,IN=,OUT1=,PRLD=,CKPTID=,DIRCA=000,
//          PARDLI=1,STIMER=,CPUTIME=,OBA=,AGN=,IMSID=$IMSID,OPT=N
//*****
/*      CHECK THE BMP PARAMETERS, ESPECIALLY THE 'IMSID' PARAMETER
//*****
//PCM      EXEC PGM=DFSRR00,REGION=1536K,
//          PARM=(BMP,BVPCMPUF,PCMPUF$SUG,&IN,&OUT1,
//          &OPT&SPIE&TEST&DIRCA,&PRLD,&STIMER,
//          &CKPTID,

```

```

//          &PARDLI,&CPUTIME,&NBA,&OBA,&IMSID,&AGN)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=SYS1.SCEERUN,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDRER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7DC$BASE DD DSN=&DSMS..&BASE.DC,DISP=SHR
//B7D3$BASE DD DSN=&DSMS..&BASE.D3,DISP=SHR
//BVP7LB   DD DSN=&INDSV..BVPLB,DISP=SHR
//BVP7PA   DD DSN=&INDSV..BVPPA,DISP=SHR
//BVP7P1   DD DSN=&INDSV..BVPP1,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//BVP7GJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//BVP7TR   DD DSN=&INDSV..BVPTR,DISP=SHR
//BVP7WS   DD DSN=&INDSV..BVPWS,DISP=SHR
//BVP7SV   DD DSN=&INDSV..BVPVSV,DISP=SHR
//BVP7QJ   DD DSN=&INDSV..BVPQJ,DISP=SHR
//

```

## Transactions Archiving

### ARPM - Introduction

The ARPM procedure archives the valid transactions already processed by the Standard Bridge and stored in the QJ journal file. It extracts the erroneous transactions, outputs the corresponding errors and recycle them again. It reconstitutes the new QJ journal file with the transactions which have not yet been processed by the Bridge and with recycled erroneous transactions.

### EXECUTION CONDITIONS

No current generation. The QJ file must be closed to on-line use.

ABNORMAL EXECUTION

If the last step (which reconstitutes the QJ journal file) abends, you just have to re-run this step. In all other cases, whichever the cause of the error is, you can run the procedure again as it is, once you have solved the problem.

**ARPM - Input / Processing / Results**

A '\*' line with the user code and password.

OUTPUT REPORTS

This procedure outputs a list of the errors previously detected by the update procedure.

**ARPM - Description of Steps**

ANALYSIS AND PREPARATION: PCM500

This step validates the user input, analyzes the journal file and prepares the archiving and output of errors.

Code	Physical name	Type	Label
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7MB	&&ARPMMB	Input	User transaction
BVP7QJ	&INDSV..BVPQJ	Input	Standard Bridge journal
PAC7XP	&&ARPMXP	Output	Counters of the reconstituted journal
PAC7XQ	&&ARPMXQ	Output	Transactions to be recycled
PAC7XR	&&ARPMXR	Output	Erroneous transactions
PAC7XS	&&ARPMXS	Output	Transactions already processed
PAC7XT	&&ARPMXT	Output	Transactions for update
PAC7DD		Report	Authorization control

ANALYSIS AND PREPARATION: PCM510

This step archives the correct transactions already processed by the Standard Bridge automaton.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7XS	&&ARPMXS	Input	Transactions already processed
PAC7RJ	&INDSN..BVPJQ(0)	Input	Former archiving
PAC7JR	&INDSN..BVPJQ(+1)	Output	New archiving

#### OUTPUT OF ERRORS: PCM520

This step outputs the list of the errors detected during the update process by the automaton.

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7XR	&&ARPMXR	Input	Extracted erroneous transactions
PAC7ET		Report	List of errors

#### RECONSTITUTION OF THE QJ JOURNAL FILE: PCM550

This step reconstitutes the new QJ journal file with the transactions which have not yet been processed by the automaton as well as with the erroneous transactions.

Code	Physical name	Type	Label
PAC7XP	&&ARPMXP	Input	Counters
PAC7XQ	&&ARPMXQ	Input	Transactions to be processed again
PAC7JQ	&INDSN..ARPMJQ	Output	New journal

#### RECONSTITUTION OF THE QJ JOURNAL FILE: PCM560

This step reconstitutes the new QJ journal file, using as input the sequential file created by PCM550.

Code	Physical name	Type	Label
PAC7JQ	&INDSN..ARPMJQ	Input	Sequential file
PAC7QJ	&INDSV..BVPQJ	Output	New journal

## ARPM - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*      - ARCHIVAL OF THE GENERIC BRIDGE JOURNAL -
/*
/*
/* -----
/* INPUT      : USER IDENTIFICATION
/* COL 2      : "*"
/* COL 3      : USER CODE
/* COL 11     : PASSWORD
/* -----
//BVPARPM  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          INDUV='$INDUV',                INDEX OF USER VSAM FILES
//          INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
//          INDSN='$INDSN',                INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//          SORTLIB='$BIBT',                SORT LIBRARY
//          DSCB='$DSCB',                  DSCB MODEL FILE
//          OUT=$OUT,                       OUTPUT CLASS
//          VOLS='SER=$VOLUN',              VOL. OF GENER.BRIDGE ARCH.JOURNAL
//          UNITS=$UNITUN,                  BACKUP UNIT (DISK OR CARTRIDGE)
//          UWK=$UWK,                       WORK UNIT
//          SPAQJ='(TRK,(50,10),RLSE)',     ARCHIVAL JOURNAL SPACE
//          SPAWS='(TRK,(30,10),RLSE)',     WORK FILE SPACE
//          PSBLIB='$PSBLIB',               LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',               LIBRARY OF DBD'S
//          RESLIB='$RESLIB',               IMS RESLIB
//          PROCLIB='$PRCLIB',              IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&ARPMMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR

```



```

//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7QJ DD DSN=&INDSV..BVPQJ,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFQJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PCMS500 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPCM500,PCM500$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//BVP7QJ DD DSN=&INDSV..BVPQJ,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&ARPM MB,DISP=(OLD,DELETE,DELETE)
//PAC7XP DD DSN=&&ARPMXP,UNIT=&UWK,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=1105,BLKSIZE=11050),
// SPACE=&SPAWS
//PAC7XQ DD DSN=&&ARPMXQ,UNIT=&UWK,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=1105,BLKSIZE=11050),
// SPACE=&SPAWS
//PAC7XR DD DSN=&&ARPMXR,UNIT=&UWK,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=1105,BLKSIZE=11050),

```

```

//          SPACE=&SPAWS
//PAC7XS DD DSN=&&ARPMXS,UNIT=&UWK,DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=1105,BLKSIZE=11050),
//          SPACE=&SPAWS
//PAC7XT DD DSN=&&ARPMXT,UNIT=&UWK,DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=1105,BLKSIZE=11050),
//          SPACE=&SPAWS
//*-----
//PCM510 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPCM510,PCM510$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:          DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7XS DD DSN=&&ARPMXS,DISP=(OLD,DELETE,DELETE)
//PAC7RJ DD DSN=&INDSN..BVPJQ(0),DISP=OLD
//PAC7JR DD DSN=&INDSN..BVPJQ(+1),
//          DISP=(,CATLG,DELETE),
//          UNIT=&UNITS,
//          VOL=&VOLS,
//          SPACE=&SPAQJ,
//          DCB=(&DCB,RECFM=FB,LRECL=1119,BLKSIZE=11190)
//*-----
//PCM520 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPCM520,PCM520$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR

```

```

//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSOUX  DD SYSOUT=&OUT
//DDSNAP  DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//        DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//        BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//        BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON  DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(3,1),,CONTIG)
//BVP7AE  DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7XR  DD DSN=&&ARPMXR,DISP=(OLD,DELETE,DELETE)
//PAC7ET  DD SYSOUT=&OUT
//*-----
//PCM550 EXEC PGM=BVPCM550,COND=(0,NE,PCM500)
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:      DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$BCOB,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSOUX  DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7XP  DD DSN=&&ARPMXP,DISP=(OLD,DELETE,DELETE)
//PAC7XQ  DD DSN=&&ARPMXQ,DISP=(OLD,DELETE,DELETE)
//PAC7JQ  DD DSN=&INDSN..ARPMJQ,UNIT=&UWK,DISP=(,PASS),
//        DCB=(RECFM=FB,LRECL=1105,BLKSIZE=11050),
//        SPACE=&SPAQJ
//*-----
//OSAMQJ1 EXEC PGM=IDCAMS
//SYSIN   DD DSN=&INDSV..BVPSY(DLBVPQJ),DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//*
//OSAMQJ2 EXEC PGM=IEFBR14
//BVP7QJ  DD DSN=&INDSV..BVP7QJ,UNIT=&UNITS,
//        DISP=(,CATLG,DELETE),VOL=SER=&VOL,
//        DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//        SPACE=&SPAQJ
//*-----
//PCM560 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//        PARM=(DLI,BVPCM560,PCM560$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),COND=((0,NE,PCM500),(0,NE,PCM550))
//STEPLIB DD DSN=&RESLIB,DISP=SHR

```

```

//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7JQ   DD DSN=&INDSN..ARPMJQ,DISP=SHR
//BVP7QJ   DD DSN=&INDSV..BVPQJ,DISP=SHR
//*-----
//IEF14    EXEC PGM=IEFBR14,COND=(0,NE,PCM560)
//DD1      DD DSN=*.PCM560.PAC7JQ,DISP=(SHR,DELETE,DELETE)
//

```

---

## 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 between two sessions or more.

Pac/Transfer scans the development database archived Journal file and reads a dedicated Parameter file.

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

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

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

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

### FUNCTIONALITIES

Pac/Transfer is used to transfer updates made in a source session to one or several target sessions.

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

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

There are no constraints regarding the chronological order of sessions. It is possible to transfer the transactions entered in a given source session into a later target session (target-session number greater than that of the source session), just as it is possible to transfer it onto a previous target session (target-session number smaller than that of the source session).

The transfer parameters for all development Databases managed by the administrator is stored in administration Database. The list of these databases is itself defined in the administration Database.

Thus, the development Database notion is now important to parameterize Pac/Transfer.

You need to have defined a logical database code for each development Database.

The logical database code used is the one entered when you restore the development database with the REST procedure execution.

In batch Pac/Transfer processing procedures, it is not necessary to indicate the logical code of the development Database. The code indicated in the data file of the database processed will be systematically taken into account. This code will be used as a link between the development database and the transfer sets stored in the administration Database throughout the processing, as indicated above.

## **Processes Chronology**

1. Updating the transfer parameters (TRUP)

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
5. Generating the transfer transactions
6. Updating the development Database
7. Reinitializing the DSMS environment

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

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

## **Update of Transfer Parameters**

### **TRUP - Introduction**

PacTransfer processing is based on user-defined parameters stored in the administration database. These parameters monitor all PacTransfer procedures.

These parameters must be created -- via a TRUP execution -- prior to any PacTransfer 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. A transaction sets is linked to a development database. The parameter file can therefore store several Transaction Sets.

During the TRUP procedure execution, an access to the development database checks the transaction sets parameters and then these parameters are stored in the administration database.

The transaction sets identifier stored in this database is made of the 'development database logical code' + 'transaction sets code'. The development database logical code is automatically affected in the identifier during the TRUP procedure processing.

The other batch procedures of Pac/Transfer use the same principle, searching the processing parameters, in the administration database, with an identifier made of the development database logical code + the transaction sets code.

By defining several Transaction Sets, you obtain flexible transfer operations, fully adaptable 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

- Transfer set header line

It is required at the beginning of the transactions relating to transfer set. It identifies the transfer set to which the parameters entered on the continuation lines are related.

- Session number

It is required to 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.

- Library

By default, ALL Libraries in the development 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.

- 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.

- DSMS change number

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

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

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

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

**NOTE:** This option overrides any target user selection such as described above.

### EXECUTION CONDITION

Since this procedure updates the network, the GR, GN, GJ and GY files must be closed to on-line access (except on hardware which supports batch/on-line concurrency).

### PRINTED REPORT

Printout of the parameter-file contents.

### **TRUP - User Input**

User identification line (required)

Position	Length	Value	Meaning
2	1	'**'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

Transfer set header line (required)

This line must precede the update transactions of a transfer set. It identifies the transfer set to which the following transactions are related.

Position	Length	Value	Meaning
1	1		Action code
		'C'	Creation
		'M'	Modification
		'D'	Deletion
		' '	Creation or modification depending on the database status



Position	Length	Value	Meaning
		'X'	Equal to ' ', without any uppercase conversion of the label
2	2	'GA'	Line type
4	10		Transfer set code (required) different from 999999999 and *****
14	36		Transfer set label (required in creation mode)

### Session selection line

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

Position	Length	Value	Meaning
1	1		Action code
		'C'	Creation
		'M'	Modification
		'D'	Deletion
		' '	Creation or modification depending on the database state
		'X'	Equal to ' ', with no uppercase conversion of the label
2	2	'GS'	Line type
4	4		Source session (required)
13	2		Line number (two lines only are authorized)
		'00'	First line for the 9 first target sessions (default value)
		'01'	Continuation line for the 9 following target sessions, if required (the target session number is limited to 18: the input in position 1 to 7 on the first line must be repeated on continuation line).
15	3		Sequence number of reports (required and numeric)
18	36		List of target sessions: The sessions are entered without 'T' and are not followed by blanks (one session is required at least)

### Selection line of Libraries

Position	Length	Value	Meaning
1	1		Action code:
		'C'	Creation
		'M'	Modification
		'D'	Deletion
		' '	Creation or modification depending on the database state
		'X'	equal to ' ', with no uppercase conversion of the label
2	2	'GB'	Line type
4	3		Source Library (required)
13	60		Library codes (20 max.) Default: source Library. Library codes are not separated by blanks.

### Selection line of user codes

Position	Length	Value	Meaning
1	1		Action code
		'C'	Creation
		'M'	Modification
		'D'	Deletion
		' '	Creation or modification depending the database state
		'X'	Equal to ' ', without any uppercase conversion of the label
2	2	'GU'	Line type
4	8		Source user code (required)
13	8		Target user code Default: source user code

### Selection line of DSMS changes

Position	Length	Value	Meaning
1	1		Action code:
		'C'	Creation

Position	Length	Value	Meaning
		'M'	Modification
		'D'	Deletion
		' '	Creation or modification depending the database state
		'X'	Equal to ' ', without any uppercase conversion of the label
2	2	'GC'	Line type
4	3		Source product code (required) it must be left-justified
7	6		Source Change number (required)
13	3		Target product code must be left-justified
16	6		Target Change number Default: Source product/Change
22	8		Target user code Default: Source user

#### Request line for multiple deletions

Multiple deletions may be requested at many levels:

- at the level of each type of selection
- for a given Transaction Set,
- at the level of the whole Set.

Position	Length	Value	Meaning
1	1	'B'	Multiple deletion request
2	2	'GA'	Deletion of whole Set
		'GS'	Deletion of set lines: 'GS' 'GB' 'GC' and 'GU'
		'GB'	Deletion of 'GB' Set line
		'GU'	Deletion of 'GU' Set line
		'GC'	Deletion of 'GC' Set line
4	10		Only if value 'GA' in columns 2 and 3.
			Set code
		/'*****'	Deletion of the whole Sets in a development Database

**TRUP - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

UPDATE OF THE ADMINISTRATION DATABASE: PTUG20

This step updates the administration Database in order to store selection parameters.

Code	Physical name	Type	Label
B7AR\$BASE	&INDUV.&BASE.AR	Input	Development database data file
B7AN\$BASE	&INDUV.&BASE.AN	Input	Development database index file
BVP7GU	&INDSV..BVPGU	Input	Users file
BVP7AE	&INDSV..BVPAE	Input	Error messages file
BVP7GR	&INDSV..BVPGR	Input/Output	Administration Database data file
BVP7GN	&INDSV..BVPGN	Input/Output	Administration Database index file
BVP7GY	&INDSV..BVPGY	Input/Output	Administration Database extension data file
BVP7GJ	&INDSV..BVPGJ	Input/Output	Administration Database Journal file
PAC7MC	&&TRUPMB	Input	Parameters update transactions file
PAC7ME	&&PAC7ME	Input/Output	Work file
PAC7MY	&&PAC7MY	Input/Output	Work file
PAC7TB	&&PAC7TB	Input/Output	Work file
PAC7BM	&&PAC7BM	Output	Parameters print request file
PAC7ET		Report	Input validation
PAC7IE		Report	Administration Database update report
PAC7IF		Report	Administration Database update errors
PAC7DD		Report	Output report: check of line '*' in relation to the development Database
PAC7DE		Report	Output report: check of line '*' in relation to the administration Database
SORTWK01		Sort	

Code	Physical name	Type	Label
SORTWK02		Sort	
SORTWK03		Sort	

### EXTRACTION OF THE ADMINISTRATION DATABASE: PTUG30

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
BVP8GN	&INDSV..BVP GN	Input	Administration Database index
BVP8GR	&INDSV..BVP GR	Input	Administration Database data
BVP8GY	DUMMY	Input	Administration Database extension data
PAC7MB	&&PAC7BM	Input	Extraction request file
PAC7GL	&&PAC7GL	Output	Target sessions list
PAC7UY	&&PAC7UY	Output	Limited parameters file
PAC7DD		Report	'*' line validation report
PAC7TK	&&PAC7TK	Input/Output	Work file
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### PRINTING OF SELECTION PARAMETERS: PTUG31

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages file
PAC7UY	&&PAC7UY	Input	Limited parameters file
BVP8GR	&INDSV..BVP GR	Input	Administration Database data file
PAC7ET		Report	Printing of the selection parameters list

## PRINTING OF THE TARGET SESSIONS FILE: PTUG32

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP7AE	Input	Error messages
BVP8GR	&INDSV..BVP8GR	Input	Administration Database data
PAC7GL	&&PAC7GL	Input	Target session
PAC7ET		Report	Printing of the target session list
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### TRUP - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      PAC/TRANSFER:
/**      UPDATE OF THE TRANSFER PARAMETERS
/** -----
/**
/** PAC/TRANSFER - PROCESSING IS BASED ON THE USER-DEFINED
/** PARAMETERS STORED IN THE ADMINISTRATION DATABASE.
/** THESE PARAMETERS CONTROL THE VARIOUS PROCESSES OF THE
/** PROCEDURES OF FACILITY.
/**
/** -----
/**
/**BVPTRUP PROC BASE=$BASE,                CODE OF VAPAC DATABASE
/**  INDUV='$INDUV',                        INDEX OF VSAM USER FILE
/**  INDSV='$INDSV',                        INDEX OF VSAM SYSTEM FILE
/**  INDSN='$INDSN',                        INDEX OF NON VSAM SYSTEM FILE
/**  OUT=$OUT,                               OUTPUT CLASS
/***:  VSAMCAT='$VCAT',                      USER VSAM CATALOG
/***:  SYSCAT='$SCAT',                       SYSTEM VSAM CATALOG
/**  STEPLIB='$HLQ..SBVPMBR8',              LOAD-MODULES LIBRARY
/**  SORTLIB='$BIBT',                        SORT LIBRARY
/**  CYL=5,                                  SORTWORK SPACE
/**  UWK=$UWK,                               WORK UNIT
/**  PSBLIB='$PSBLIB',                       LIBRARY OF PSB'S
/**  DBDLIB='$DBDLIB',                       LIBRARY OF DBD'S
/**  RESLIB='$RESLIB',                       IMS RESLIB
/**  PROCLIB='$PRCLIB',                      IMS PROCLIB
/**  BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
/**  CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
/*******
/**INPUT  EXEC PGM=BVPTU001
/**STEPLIB DD DSN=&STEPLIB,DISP=SHR
/**        DD DSN=$BCOB,DISP=SHR
/**CARTE  DD DDNAME=SYSIN,DCB=BLKSIZE=80

```

```

//PAC7MB DD DSN=&&TRUPMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGJ DD DSN=&INDSV..BVPGJ,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PTUG20 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG20,PTUG20$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//BVP7GJ DD DSN=&INDSV..BVP7GJ,DISP=SHR

```

```

//BVP7GY DD DSN=&INDSV..BVPGY,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7MY DD DSN=&&PAC7MY,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7ME DD DSN=&&PAC7ME,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=170,BLKSIZE=17000)
//PAC7TB DD DSN=&&PAC7TB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=1028,BLKSIZE=10280)
//PAC7BM DD DSN=&&PAC7BM,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7IE DD SYSOUT=&OUT
//PAC7IF DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7DE DD SYSOUT=&OUT
//PAC7ET DD SYSOUT=&OUT
//PAC7MC DD DSN=&&TRUPMB,DISP=(OLD,DELETE)
//*-----
//PTUG30 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG30,GTUG30$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=(0,NE,PTUG20)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR

```



```

//BVP8GY DD DUMMY
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7MB DD DSN=&&PAC7BM,DISP=(OLD,DELETE)
//PAC7GL DD DSN=&&PAC7GL,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7TK DD DSN=&&PAC7TK,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=1028,BLKSIZE=10280)
//PAC7UY DD DSN=&&PAC7UY,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=180,BLKSIZE=18000)
//PAC7DD DD SYSOUT=&OUT
//*-----
//PTUG31 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG31,GTUG31$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=((0,NE,PTUG20),(0,NE,PTUG30))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//PAC7UY DD DSN=&&PAC7UY,DISP=(OLD,DELETE)
//PAC7ET DD SYSOUT=&OUT
//*-----
//PTUG32 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG32,GTUG32$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM),
// COND=((0,NE,PTUG20),(0,NE,PTUG30))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR

```

```

//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVP8GR,DISP=SHR
//PAC7GL   DD DSN=&&PAC7GL,DISP=(OLD,DELETE)
//PAC7ET   DD SYSOUT=&OUT
//*
```

## Print of Transfer Parameters

### TRED - Introduction

This procedure is used to print all the parameters for each development Database and transfer set.

It is possible to print all the parameters, or to print those of only one development Database.

### TRED - User Input

User identification line (required).

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
29	4		Selection of the development Database to be printed
		bbbb	Selection of a Database
		'****'	Selection of all Databases

**TRED - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

CHECKING OF THE PROCESS REQUEST: PTUG28

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GR	&INDSV..BVPGR	Input	Administration database data
BVP7GN	&INDSV..BVPGN	Input	Administration database index
BVP7GU	&INDSV..BVPGU	Input	Administration database users
BVP8GR	&INDSV..BVPGR	Input	Administration database data
PAC7MB	&&TREDMB	Input	Parameters print request
PAC7BM	&&PAC7BM	Output	Parameters print request
PAC7DD		Report	Output report: user code validity check
PAC7ET		Report	Output report: print request check

EXTRACTION OF THE ADMINISTRATION DATABASE: PTUG30

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
BVP8GY	DUMMY	Input	Administration Database extension data
PAC7MB	&&PAC7BM	Input	Printing request file
PAC7GL	&&PAC7GL	Output	Target sessions file
PAC7UY	&&PAC7UY	Output	Limited parameters file
PAC7TK	&&PAC7TK	Input/Output	Working file
PAC7DD		Report	'*' line validation report
SORTWK01		Sort	
SORTWK02		Sort	

Code	Physical name	Type	Label
SORTWK03		Sort	

### PRINTING OF SELECTION PARAMETERS: PTUG31

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7UY	&&PAC7UY	Input	Limited parameters
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
PAC7ET		Report	List of selection parameters

### PRINTING OF TARGET SESSIONS LIST: PTUG32

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7GL	&&PAC7GL	Input	Target sessions
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
PAC7ET		Report	List of target sessions
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### **TRED - Execution JCL**

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      PAC/TRANSFER -
/**      EDITING THE DATABASE PARAMETERS
/** -----
/**
/** FOR ALL THE DATABASE OR ONE DATABASE
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/**
/** -----
/**
/**BVPTRD PROC OUT=$OUT,          OUPUT CLASS
/**  INDSN='$INDSN',              INDEX OF NON VSAM USER FILE
/**  INDSV='$INDSV',              INDEX OF VSAM SYSTEM FILE
/***: VSAMCAT='$VCAT',           USER VSAM CATALOG

```

```

//*: SYSCAT='$SCAT',          SYSTEM VSAM CATALOG
// STEPLIB='$HLQ..SBVPMBR8',  LOAD-MODULES LIBRARY
// SORTLIB='$BIBT',          SORT LIBRARY
// CYL=5,                     SORTWORK SPACE
// UWK=$UWK,                  WORK UNIT
// PSBLIB='$PSBLIB',         LIBRARY OF PSB'S
// DBDLIB='$DBDLIB',         LIBRARY OF DBD'S
// RESLIB='$RESLIB',         IMS RESLIB
// PROCLIB='$PRCLIB',        IMS PROCLIB
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&TREDMB,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PTUG28 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG28,GTUG28$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)

```

```

//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INSDN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7MB DD DSN=&&TREDMB,DISP=(OLD,DELETE)
//PAC7DD DD SYSOUT=&OUT
//PAC7ET DD SYSOUT=&OUT
//PAC7BM DD DSN=&&PAC7BM,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//*-----
//PTUG30 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG30,GTUG30$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM),
// COND=(0,NE,PTUG28)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INSDN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GY DD DUMMY
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7MB DD DSN=&&PAC7BM,DISP=(OLD,DELETE)
//PAC7GL DD DSN=&&PAC7GL,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),

```

```

//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7TK DD DSN=&&PAC7TK,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=1028,BLKSIZE=10280)
//PAC7UY DD DSN=&&PAC7UY,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=180,BLKSIZE=18000)
//PAC7DD DD SYSOUT=&OUT
//*-----
//PTUG31 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG31,GTUG31$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM),
//          COND=((0,NE,PTUG28),(0,NE,PTUG30))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//PAC7UY DD DSN=&&PAC7UY,DISP=(OLD,DELETE)
//PAC7ET DD SYSOUT=&OUT
//*-----
//PTUG32 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG32,GTUG32$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM),
//          COND=((0,NE,PTUG28),(0,NE,PTUG30))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR

```

```

//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//PAC7GL DD DSN=&&PAC7GL,DISP=(OLD,DELETE)
//PAC7ET DD SYSOUT=&OUT
//*
```

## Compression of Archived Journal

### TRJC - Introduction

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

User input may include an interval of dates and/or session numbers in order to limit transfer processing to the archived Journal's transactions belonging to that interval only.

If there is no optional user input, the compression is carried out on the complete archived Journal.

You also 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.

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

### EXECUTION CONDITIONS

None.



## RESULT

A smaller archived Journal including 'useful' transactions only.

## PRINTED REPORT

Statistical data on the TRJC execution.

### **TRJC - User Input**

User identification line (required)

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

### Options

Position	Length	Value	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

### **TRJC - Description of Steps**

INPUT RECOGNITION: PTU001

DELETE/DEFINE OF COMPRESSED JOURNAL FILE: IDCAMS

COMPRESSION (FIRST PHASE): PTUG04

Code	Physical Name	Type	Label
BVP7AE	&INDSN..BVPAE	Input	Error messages
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data

Code	Physical Name	Type	Label
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7MB	&&TRJCMB	Input	User input
PAC7BM	&&TRJCBM	Output	User input
PAC7DD		Report	Batch procedures' execution report

COMPRESSION (FIRST PHASE): PTUG05

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7PJ	&INDUN..&BASE.PJ(0)	Input	Sequential image of the development Database
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
PAC7MB	&&TRJCBM	Input	User Input
PAC7GP	&&PAC7GP	Output	Temporary journal
PAC7ET		Report	Input validation
PAC7DD		Report	Batch procedures error status
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

COMPRESSION (SECOND PHASE): PTUG06

Code	Physical name	Type	Label
PAC7GP	&&PAC7GP	Input	Temporary Journal
PAC7PK	&&PAC7PK	Output	Compressed sequential journal
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

## CLASSIFICATION OF DELETIONS/CREATIONS: PTUG07

Code	Physical name	Type	Label
B7AN\$BASE	&INDUV.&BASE.AN	Input	Development Database index
PAC7PK	&&PAC7PK	Input	Temporary journal
PAC7PL	&INDUN.&BASE.JT	Output	Compressed sequential journal
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### TRJC - Execution JCL

```

/** -----
/**          VISUALAGE PACBASE
/**
/** -----
/**          PAC/TRANSFER -
/**          COMPRESSION OF ARCHIVED JOURNAL
/** -----
/**
/**
/** FROM THE DATABASE ARCHIVED JOURNAL, THE TRJC
/** PROCEDURE PRODUCES A COMPRESSED JOURNAL
/** CONTAINING ONLY USEFUL TRANSACTIONS,
/** BY ELIMINATING THE INTERMEDIARY TRANSACTIONS
/** WHICH ARE KNOWN TO BE USELESS FOR THE TRANSFER.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/** - COMMAND LINE :
/** COL 1  : DELETION OF USER CODES:
/**          "0"  YES
/**          "1"  NO
/** COL 2  : DELETION OF CHANGE NUMBERS:
/**          "0"  YES
/**          "1"  NO
/** COL 3  : (4 CAR.) START SESSION NUMBER
/** COL 7  : (4 CAR.) END SESSION NUMBER
/**
/** COL 11 : (8 CAR.) START DATE IN THE FORM CCYYMMDD
/** COL 19 : (8 CAR.) END DATE IN THE FORM CCYYMMDD
/** -----
/**
/**/BVPTRJC PROC BASE=$BASE,          CODE OF VAPAC DATABASE
/**  INDUV='$INDUV',                  INDEX OF VSAM USER FILES
/**  INDSV='$INDSV',                  INDEX OF VSAM SYSTEM FILES
/**  INDSN='$INDSN',                  INDEX OF NON VSAM SYSTEM FILES
/**  INDUN='$INDUN',                  INDEX OF NON-VSAM USER FILES
/**  SPAJT='(TRK,(60,15),RLSE)',      COMPRESSED JOURNAL SPACE
/**  SPAPJ='(TRK,(150,15),RLSE)',    TEMPORARY FILE SPACE
/**  UNITS=$UNITN,                    COMPRESSED JOURNAL FILE

```

```

// VOLS='SER=$VOLUN',          COMPRESSED JOURNAL FILE VOLUME
// OUT=$OUT,                   OUTPUT CLASS
//*: VSAMCAT='$VCAT',          VSAM USER CATALOG
//*: SYSTCAT='$SCAT',          VSAM SYSTEM CATALOG
// STEPLIB='$HLQ..SBVPMBR8',   LOAD-MODULES LIBRARY
// SORTLIB='$BIBT',            SORT LIBRARY
// CYL=(10,2),                 SORTWORK SIZE
// UWK=$UWK,                   WORK UNIT
// PSBLIB='$PSBLIB',           LIBRARY OF PSB'S
// DBDLIB='$DBDLIB',           LIBRARY OF DBD'S
// RESLIB='$RESLIB',           IMS RESLIB
// PROCLIB='$PRCLIB',          IMS PROCLIB
// BUF=40, SPIE=0, TEST=0, EXCPVR=0, RST=0, PRLD=, SRCH=0,
// CKPTID=, MON=N, LOGA=0, FMTO=T, DBRC=$DBRC, IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&TRJCMB,DISP=(,PASS),UNIT=&UWK,
// SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//*-----
//DELJT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.JT),DISP=SHR
//*-----
//PTUG04 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG04,PTUG04$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSN..BVPAE,DISP=SHR

```

```

//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7BM DD DSN=&&TRJCBM,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7DD DD SYSOUT=&OUT
//PAC7MB DD DSN=&&TRJCMB,DISP=(OLD,DELETE)
//*-----
//PTUG05 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG05,PTUG05$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSN..BVP7AE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7PJ DD DSN=&INDUN..&BASE.PJ(0),DISP=OLD
//PAC7MB DD DSN=&&TRJCBM,DISP=(OLD,DELETE)
//PAC7ET DD SYSOUT=&OUT
//PAC7DD DD SYSOUT=&OUT
//PAC7GP DD DSN=&&PAC7GP,DISP=(,PASS),UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=233,BLKSIZE=27960),
// SPACE=&SPAPJ
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//*-----
//PTUG06 EXEC PGM=BVPTUG06,COND=(0,NE,PTUG05)
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR

```

```

//*:          DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//PAC7GP    DD DSN=&&PAC7GP,DISP=(OLD,PASS)
//PAC7PK    DD DSN=&&PAC7PK,DISP=(,PASS),UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=170,BLKSIZE=6290),
//          SPACE=&SPAPJ
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//SYSPRINT  DD SYSOUT=&OUT
//SYSUDUMP  DD SYSOUT=&OUT
//*-----
//PTUG07    EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG07,PTUG07$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=(0,NE,PTUG05)
//STEPLIB   DD DSN=&RESLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//DFSRESLB  DD DSN=&RESLIB,DISP=SHR
//IMS       DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT  DD SYSOUT=&OUT
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//DDSNAP    DD SYSOUT=&OUT
//PROCLIB   DD DSN=&PROCLIB,DISP=SHR
//IEFRDER   DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP  DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP  DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON    DD DUMMY
//DFSVSAMP  DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7PK    DD DSN=&&PAC7PK,DISP=(OLD,PASS)
//PAC7PL    DD DSN=&INDUN..&BASE.JT,
//          DISP=(,CATLG),
//          UNIT=&UNITS,VOL=&VOLS,
//          DCB=(RECFM=FB,LRECL=170,BLKSIZE=6290),
//          SPACE=&SPAJT

```

## Creation of the Transfer File

### 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:

- The only transactions processed are those meeting the source selection parameters (sessions, Libraries, users, Changes),
- The values of the selected parameters are replaced by those of the target parameters specified in the Parameter file,
- 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 relating to a development database.

### EXECUTION CONDITIONS

None.

### RESULT

The TRPF procedure produces a Transfer file and a Parameter file, reduced and adapted to the processing request (UY). These two files will be used by the TRRP procedure.

### TRPF - User Input

User identification line (required)

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

Transaction Set for processing selection line (required)

Position	Length	Value	Meaning
2	2	'LT'	
4	10	llllllllll	Code of Transaction Set to process
		*****	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 13).

**TRPF - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

PROCESSING REQUEST CHECK: PTUG27

Code	Physical name	Type	Label
BVP7AE	&INDSV.BVPAE	Input	Error messages
BVP7GR	&INDSV.BVPGR	Input	Administration Database data
BVP7GN	&INDSV.BVPGN	Input	Administration Database index
BVP7GU	&INDSV.BVPGU	Input	Administration Database users
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7MB	&&TRPFMB	Input	Parameters extraction request
PAC7BM	&&PAC7BM	Output	Parameters extraction request
PAC7DD		Report	Output report: user code validity check
PAC7ET		Report	Output report: extraction request check

EXTRACTION OF THE ADMINISTRATION DATABASE: PTUG30

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
BVP8GY	DUMMY	Input	Administration Database extension data
PAC7MB	&&PAC7BM	Input	Extraction request
PAC7GL	&&PAC7GL	Output	List of target sessions



Code	Physical name	Type	Label
PAC7UY	&&PAC7UY	Output	Limited parameters for TRRP processing
PAC7DD		Report	'*' line validation report
PAC7TK	&&PAC7TK	Output	Work file
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### PREPARATION OF LIMITED PARAMETERS FILE: BVPRMSYS

This program is used to parameterize the name of KSDS working file. The DELETE/DEFINE of the file is in member DFSYTRPF of SY PDS parameters. It is suffixed with Dsname &USER parameter. The PRMSYS program valorizes this parameter with the value entered at the execution time. The purpose is to allocate different working files from an execution to another one and to avoid any conflict when allocating working files in case of simultaneous executions.

This file will be used by the TRRP procedure

Code	Physical name	Type	Label
PACRIN	&INDSN..BVPSY(DFSYTRPF)	Input	File DEFINE skeleton
PACROU	&&DFSYTRPF	Output	DEFINE file (SYSIN for IDCAMS)

#### DEFINITION OF LIMITED PARAMETERS FILE: IDCAMS

Code	Physical name	Type	Label
SYSIN	&&DFSYTRPF	Input	DEFINE file

#### CREATION OF LIMITED PARAMETERS FILE: IDCAMS REPRO

Code	Physical name	Type	Label
IN1	&&PAC7UY	Input	Parameters
OU1	&INDUV..SYTRPF.&USER	Output	Parameters

## DELETE OF THE TRANSFERS FILE: IDCAMS

Code	Physical name	Type	Label
SYSIN	&INDUN..&BASE.SY(DL&BASE.TJ)	Input	Transfers file

## CREATION OF TRANSFERS FILE: PTUG50

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database index
BVP7GR	&INDSV..BVP GR	Input	Administration Database data
BVP7GU	&INDSV..BVP GU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7JT	&INDUN..&BASE.PJ(0)	Input	Sequential or compressed Journal
PAC7MB	&&PAC7BM	Input	User input
PAC7UY	&&PAC7UY	Input	Limited parameters
PAC7TJ	&INDUN..&BASE.TJ	Output	Transfers file
PAC7DD		Report	User line validation report
PAC7ER		Report	List of user input
PAC7ET		Report	Statistics on transfers
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

### **TRPF - Execution JCL**

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      PAC/TRANSFER -
/**      CREATING THE TRANSFER FILE
/** -----
/**
/** FROM THE ARCHIVED JOURNAL THE TRPF PROCEDURE PRODUCES
/** A TRANSFER FILE.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/** - COMMAND LINE :
/** COL 2  : "LT"
/** COL 4  : (10 CAR.) SELECTED TRANSACTION SET CODE.
```

```

//*          IF SELECTION OF ALL SETS "*****"
//*
//* -----
//*
//BVPTRPF PROC BASE=$BASE,          CODE OF VAPAC DATABASE
//  INDUV='$INDUV',                INDEX OF VSAM USER FILES
//  INDSV='$INDSV',                INDEX OF VSAM SYSTEM FILES
//  INDUN='$INDUN',                INDEX OF USER NON-VSAM FILE (TJ)
//  INDSN='$INDSN',                INDEX OF SYSTEM NON-VSAM FILE
//  SPATJ='(TRK,(60,15),RLSE)',    JOURNAL TO REPORT
//  UNITS=$UNITUN,                 NON-VSAM USER FILE (TJ)
//  VOLS='SER=$VOLUN',             NUMBER OF NON-VSAM USER FILE (TJ)
//  OUT=$OUT,                       OUTPUT CLASS
//*: VSAMCAT='$VCAT',              VSAM USER CATALOG
//*: SYSCAT='$SCAT',              VSAM SYSTEM CATALOG
//  STEPLIB='$HLQ..SBVPMBR8',      LOAD-MODULES LIBRARY
//  SORTLIB='$BIBT',              SORT LIBRARY
//  USER=,                          USER CODE
//  CYL=5,                           SORTWORK SPACE
//  UWK=$UWK,                         WORK UNIT
//  PSBLIB='$PSBLIB',              LIBRARY OF PSB'S
//  DBDLIB='$DBDLIB',              LIBRARY OF DBD'S
//  RESLIB='$RESLIB',              IMS RESLIB
//  PROCLIB='$PRCLIB',             IMS PROCLIB
//  BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//  CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&TRPFMB,DISP=(,PASS),UNIT=&UWK,
// SPACE=(TRK,1),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PTUG27 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG27,PTUG27$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM)

```

```

//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7BM DD DSN=&&PAC7BM,DISP=(,PASS),
// UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7DD DD SYSOUT=&OUT
//PAC7ET DD SYSOUT=&OUT
//PAC7MB DD DSN=&&TRPFMB,DISP=(OLD,DELETE)
//*-----
//PTUG30 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTUG30,GTUG30$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM),
// COND=(0,NE,PTUG27)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//*:STEPCHAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)

```

```

//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GY   DD DUMMY
//BVP7GN   DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7MB   DD DSN=&&PAC7MB,DISP=(OLD,PASS)
//PAC7GL   DD DSN=&&PAC7GL,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7TK   DD DSN=&&PAC7TK,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=1028,BLKSIZE=10280)
//PAC7UY   DD DSN=&&PAC7UY,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=180,BLKSIZE=18000)
//PAC7DD   DD SYSOUT=&OUT
//*-----
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER'
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYTRPF),DISP=SHR
//PACROU DD DSN=&&DFSYTRPF,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----
//DEFINE EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYTRPF,DISP=(OLD,DELETE)
//*-----
//REPRO EXEC PGM=IDCAMS,COND=((0,NE,PTUG27),(0,NE,PTUG30))
//SYSPRINT DD SYSOUT=&OUT
//IN1 DD DSN=&&PAC7UY,DISP=SHR
//OU1 DD DSN=&INDUN..SYTRPF.&USER,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO),DISP=SHR
//*-----
//DELTJ EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&INDUN..&BASE.SY(DL&BASE.TJ),
//          DISP=SHR
//*-----
//PTUG50 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG50,PTUG50$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,

```

```

//      &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//IMS    DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:     DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON  DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//PAC7MB  DD DSN=&&PAC7BM,DISP=(OLD,DELETE)
//PAC7UY  DD DSN=&INDUV..SYTRPF.&USER,DISP=SHR
//PAC7ER  DD SYSOUT=&OUT
//PAC7ET  DD SYSOUT=&OUT
//PAC7DD  DD SYSOUT=&OUT
//PAC7JT  DD DSN=&INDUN..&BASE.PJ(0),DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7AE  DD DSN=&INDSV..BVP AE,DISP=SHR
//BVP7GN  DD DSN=&INDSV..BVP GN,DISP=SHR
//BVP7GR  DD DSN=&INDSV..BVP GR,DISP=SHR
//BVP7GU  DD DSN=&INDSV..BVP GU,DISP=SHR
//PAC7TJ  DD DSN=&INDUN..&BASE.TJ,
//      DISP=(,CATLG),
//      UNIT=&UNITS,
//      VOL=&VOLS,
//      DCB=(RECFM=FB,LRECL=184,BLKSIZE=18400),
//      SPACE=&SPATJ

```

## Preparing DSMS Environment

### TRDU - Introduction

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

TRDU can operate for either one or all of the Sets defined in the Parameter 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:

- 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 authorized.

This update must be executed BEFORE the transfer operation.

- 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 development database.

#### EXECUTION CONDITIONS

None.

#### RESULTS

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

### TRDU - User Input

User identification line (required)

Position	Length	Value	Meaning
2	1	'**'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password

Transaction set selection line (required)

Position	Length	Value	Meaning
2	2	'LT'	
4	10	llllllllll	Selected Transaction Set code
		*****	Selection of all Sets

One and only one LT-type line is required.

### TRDU - Description of Steps

INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

CHECK ON PROCESSING REQUEST: PTUG26

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database data
BVP7GR	&INDSV..BVP GR	Input	Administration Database index
BVP7GU	&INDSV..BVP GU	Input	Administration Database index
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7MB	&&TRDUMB	Input	Parameters extraction request
PAC7BM	&&TRDUBM01	Output	Parameters extraction request
PAC7ET		Report	Output report: check on user code validity
PAC7DD		Report	Output report: check on print request



## EXTRACTION OF THE ADMINISTRATION DATABASE: PTUG30

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
BVP8GY	DUMMY	Input	Administration Database extension data
PAC7MB	&&TRDUBM01	Input	Extraction request
PAC7GL	&&PAC7GL	Output	Target sessions
PAC7UY	&&PAC7UY	Output	Limited parameters
PAC7DD		Report	Report output: '*' line validation
PAC7TK	&&PAC7TK	Input/Output	Work file
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

## PREPARATION OF LIMITED PARAMETERS FILE: BVPRMSYS

This program is used to parameterize the name of the KSDS working file.

The DELETE/DEFINE of the file is to be found in DFSYTRDU member of SY parameters PDS. It is suffixed with Dsname &USER parameter. The PRMSYS program valorizes this parameter with the value entered at execution time.

The purpose is to allocate different working files from one execution to another one and to avoid conflicts when allocating working files during simultaneous executions.

Code	Physical name	Type	Label
PACRIN	&INDSN..BVPSY(DFSYTRDU)	Input	File DEFINE skeleton
PACROU	&&DFSYTRDU	Output	DEFINE file (SYSIN for IDCAMS)

### DEFINITION OF LIMITED PARAMETERS: IDCAMS

Code	Physical name	Type	Label
SYSIN	&&DFSYTRDU	Input	DEFINE file

### CREATION OF LIMITED PARAMETERS FILE: IDCAMS REPRO

Code	Physical name	Type	Label
IN1	&&PAC7UY	Input	
OU1	&INDUV..SYTRDU.&USER	Output	

### SELECTION OF SETS: PTUG42

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7UY	&INDUV..SYTRDU.&USER	Input	Limited parameters
PAC7MB	&&PAC7BM01	Input	User input
PAC7BM	&&PAC7BM02	Output	Sets file
PAC7DD		Report	Report output: check on users
PAC7ET		Report	Report output: check on extraction

### DSMS PREPARATION BEFORE TRANSFER: PTUG44

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7UY	&INDUV..SYTRDU.&USER	Input	Limited parameters
B7DC\$BASE	&DSMS..&BASE.DC	Input	Elements of the product (DSMS)
B7D3\$BASE	&DSMS..&BASE.D3	Input	Index of product (DSMS) elements
PAC7MB	&&PAC7BM02	Input	Batch transactions

Code	Physical name	Type	Label
PAC7CI	&&TRDUCI	Output	Source/target initial state creation transactions
PAC7SI	&&TRDUSI	Output	Source/target initial state deletion transactions
PAC7GC	&&TRDUGC	Output	Target-change authorization preparation file
PAC7ET		Report	Report

#### GENERATION OF TARGET CHANGE TRANSACTIONS: PTUG46

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7GC	&&TRDUGC	Input	Preparation file for target-change authorizations
PAC7CC	&&TRDUCC	Output	Target creation transactions before transfer
PAC7SC	&&TRDUSC	Output	Target deletion transactions after transfer
PAC7ET		Report	Report output
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

#### CREATION OF BEFORE-TRANSFER TRANSACTIONS FILE: IEBGENER

Code	Physical name	Type	Label
SYSUT1	&&TRDUSI	Input	Initial state deletion transactions
SYSUT1	&&TRDUCC	Input	Target creation transactions before transfer
SYSUT2	&&TRDUAV	Output	Transactions for DUPT before updating the development Database

## CREATION OF TRANSACTIONS FILE AFTER TRANSFER: IEBGENER

Code	Physical name	Type	Label
SYSUT1	&&TRDUSC	Input	Initial state creation transactions
SYSUT1	&&TRDUCI	Input	Target deletion transactions after transfer
SYSUT2	&&TRDUAP	Output	Transactions for DUPT after updating the development Database

## DELETE OF THE LIMITED PARAMETERS FILE: BVPRMSYS

This program is used to parameterize the name of the KSDS work file.

The DELETE/DEFINE of the file is to be found in the DFSYTRDU member of SY parameters PDS. Its Dsname is suffixed with the &USER parameter. The PRMSYS program valorizes this parameter with the value entered at execution time.

The purpose is to allocate different work files from one execution to another one and to avoid conflicts when allocating work files during simultaneous executions.

Code	Physical name	Type	Label
PACRIN	&INDSN..BVPSY(DLSYTRDU)	Input	
PACROU	&&DLSYTRDU	Output	File DEFINE (SYSIN for IDCAMS)

## DELETE OF THE LIMITED PARAMETERS FILE: IDCAMS

Code	Physical name	Type	Label
SYSIN	&&DLSYTRDU	Input	

### **TRDU - Execution JCL**

```
/** -----  
/**      VISUALAGE PACBASE  
/**  
/** -----  
/**      PAC/TRANSFER -  
/**      PREPARING THE DSMS ENVIRONMENT  
/** -----  
/**  
/** THE DSMS-ENVIRONMENT PREPARATION PROCEDURE  
/** (TRDU) MUST BE USED WHEN THE DEVELOPMENT DATABASE
```

```

/* IS UNDER DSMS CONTROL, AND WHEN SOURCE CRITERIA INCLUDE
/* A SELECTED CHANGE NUMBER.
/*
/* INPUT :
/* - USER IDENTIFICATION LINE (REQUIRED)
/* - COMMAND LINE :
/* COL 2 : "LT"
/* COL 4 : (10 CAR.) SELECTED TRANSACTION SET CODE.
/* IF SELECTION OF ALL SETS "*****"
/*
/* -----
/*
//BVPTRDU PROC BASE=$BASE,          CODE OF VAPAC DATABASE
//  INDUV='$INDUV',                INDEX OF USER VSAM FILES
//  INDSV='$INDSV',                INDEX OF SYSTEM FILES
//  INDSN='$INDSN',                INDEX OF NON VSAM SYSTEM FILES
//  DSMS='$DSMS',                  DSN OF DSMS PRODUCT ELEMENTS FILE
//  SPACI='(TRK,(90,15),RLSE)',    'CI' FILE SPACE
//  SPACC='(TRK,(90,15),RLSE)',    'CC' FILE SPACE
//  SPAGC='(TRK,(90,15),RLSE)',    'GC' FILE SPACE
//  SPAAV='(TRK,(90,15),RLSE)',    'AV' FILE SPACE
//  SPAAP='(TRK,(90,15),RLSE)',    'AP' FILE SPACE
//  USER=,                          USER CODE
//  CYL=2,                            NUMBER OF SORTWORK CYLINDERS
//  OUT=$OUT,                          OUTPUT CLASS
/**: VSAMCAT='$VCAT',                VSAM USER CATALOG
/**: SYSTCAT='$SCAT',                VSAM SYSTEM CATALOG
//  STEPLIB='$HLQ..SBVPMBR8',        LOAD-MODULES LIBRARY
//  SORTLIB='$BIBT',                SORT LIBRARY
//  UWK=$UWK,                          WORK UNIT
//  PSBLIB='$PSBLIB',                LIBRARY OF PSB'S
//  DBDLIB='$DBDLIB',                LIBRARY OF DBD'S
//  RESLIB='$RESLIB',                IMS RESLIB
//  PROCLIB='$PRCLIB',                IMS PROCLIB
//  BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//  CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//CARTE DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB DD DSN=&&TRDUMB,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//-----
//VERIFY EXEC PGM=IDCAMS
/**:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
/**: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR

```

```

//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//*-----
//PTUG26 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG26,PTUG26$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7BM   DD DSN=&&TRDUBM01,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7DD   DD SYSOUT=&OUT
//PAC7ET   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&TRDUMB,DISP=(OLD,DELETE)
//*-----
//PTUG30 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG30,GTUG30$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=(0,NE,PTUG26)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR

```

```

//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GY   DD DUMMY
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7MB   DD DSN=&&TRDUBM01,DISP=(OLD,PASS)
//PAC7GL   DD DSN=&&PAC7GL,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000)
//PAC7TK   DD DSN=&&PAC7TK,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=1028,BLKSIZE=10280)
//PAC7UY   DD DSN=&&PAC7UY,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(60,15),RLSE),
//          DCB=(RECFM=FB,LRECL=180,BLKSIZE=18000)
//PAC7DD   DD SYSOUT=&OUT
//*-----
//PRMSYS  EXEC PGM=BVPRMSYS,PARM='&USER'
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//PACRIN  DD DSN=&INDSN..BVPSY(DFSYTRDU),DISP=SHR
//PACROU  DD DSN=&&DFSYTRDU,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----
//DEFINE  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN   DD DSN=&&DFSYTRDU,DISP=(OLD,DELETE)
//*-----
//REPRO   EXEC PGM=IDCAMS,COND=((0,NE,PTUG26),(0,NE,PTUG30))
//IN1     DD DSN=&&PAC7UY,DISP=SHR
//OU1     DD DSN=&INDUV..SYTRDU.&USER,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT

```

```

//SYSIN DD DSN=&INDSN. .BVPSY(REPRO),DISP=SHR
//*-----
//PTUG42 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTUG42,PTUG42$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((0,NE,PTUG26),(0,NE,PTUG30))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN. .BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB   DD DSN=&&TRDUBM01,DISP=(OLD,DELETE)
//PAC7UY   DD DSN=&INDUV. .SYTRDU.&USER,DISP=SHR
//B7AR$BASE DD DSN=&INDUV. .&BASE.AR,DISP=SHR
//BVP7AE   DD DSN=&INDSV. .BVP AE,DISP=SHR
//BVP7GN   DD DSN=&INDSV. .BVP GN,DISP=SHR
//BVP7GR   DD DSN=&INDSV. .BVP GR,DISP=SHR
//BVP7GU   DD DSN=&INDSV. .BVP GU,DISP=SHR
//PAC7DD   DD SYSOUT=&OUT
//PAC7ET   DD SYSOUT=&OUT
//PAC7BM   DD DSN=&&TRDUBM02,DISP=(,PASS),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
//      UNIT=&UWK,SPACE=(TRK,(1,1),RLSE)
//*-----
//PTUG44 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTUG44,PTUG44$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((0,NE,PTUG26),(0,NE,PTUG30))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR

```



```

//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB DD DSN=&&TRDUBM02,DISP=(OLD,DELETE)
//PAC7UY DD DSN=&INDUV..SYTRDU.&USER,DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7DC$BASE DD DSN=&DSMS..&BASE.DC,DISP=SHR
//B7D3$BASE DD DSN=&DSMS..&BASE.D3,DISP=SHR
//PAC7ET DD SYSOUT=&OUT
//PAC7CI DD DSN=&&TRDUCI,DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=250,BLKSIZE=27500),
//          UNIT=&UWK,SPACE=&SPACI
//PAC7SI DD DSN=&&TRDUSI,DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=250,BLKSIZE=27500),
//          UNIT=&UWK,SPACE=&SPACI
//PAC7GC DD DSN=&&TRDUGC,DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=40,BLKSIZE=32000),
//          UNIT=&UWK,SPACE=&SPAGC
//*-----
//PTUG46 EXEC PGM=DFSRRCO0,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG46,PTUG46$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM),
//          COND=((0,NE,PTUG26),(0,NE,PTUG30))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR

```

```

//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1)),CONTIG)
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..$BASE.AR,DISP=SHR
//PAC7GC DD DSN=&&TRDUGC,DISP=SHR
//PAC7CC DD DSN=&&TRDUCC,DISP=(NEW,PASS),
//
// UNIT=&UWK,SPACE=&SPACC,
//
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
//PAC7SC DD DSN=&&TRDUSC,DISP=(NEW,PASS),
//
// UNIT=&UWK,SPACE=&SPACC,
//
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
//PAC7ET DD SYSOUT=&OUT
//*-----
//CREAV EXEC PGM=IEBGENER,COND=((0,NE,PTUG26),(0,NE,PTUG30))
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD DSN=&&TRDUSI,DISP=SHR
//
// DD DSN=&&TRDUCC,DISP=SHR
//SYSUT2 DD DSN=&&TRDUAV,DISP=(NEW,PASS),
//
// UNIT=&UWK,SPACE=&SPAAP,
//
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
//*-----
//CREAP EXEC PGM=IEBGENER,COND=((0,NE,PTUG26),(0,NE,PTUG30))
//SYSOUT DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DUMMY
//SYSUT1 DD DSN=&&TRDUSC,DISP=SHR
//
// DD DSN=&&TRDUCI,DISP=SHR
//SYSUT2 DD DSN=&&TRDUAP,DISP=(NEW,PASS),
//
// UNIT=&UWK,SPACE=&SPAAP,
//
// DCB=(RECFM=FB,LRECL=250,BLKSIZE=2500)
//*-----
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER',COND=EVEN
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYTRDU),DISP=SHR
//PACROU DD DSN=&&DLSYTRDU,DISP=(,PASS),SPACE=(TRK,1),
//
// UNIT=&UWK,
//
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYTRDU,DISP=(OLD,DELETE)

```

## DSMS Update Before Database Update

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

## TRRP - Generation of Transfer Transactions

### TRRP - Introduction

Once the transfer file has been built, the TRRP procedure generates transfer transactions. These have the same format as batch update transactions applicable in the development Database 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.

#### STANDARD EXECUTION (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 (UY) once all transactions have been generated for a given session.

As a result, if this attribute is positioned for a given session (see also the other execution modes, described in Paragraphs 2 and 3), transactions for that session will not be generated and TRRP will automatically proceed with the next target session, as listed in the Parameter file.

This execution mode brings an automatic control over your transfer operations since it avoids duplicating transactions which could otherwise happen when prior TRRP executions have been run.

The TRRP standard execution mode is therefore recommended for sites where PacTransfer operations involve large volumes of transactions.

A Warning message will tell you when all sessions have been dealt with.

Generated transactions must then be used by the batch update procedure (UPDT) of the development database.

You may prefer to concatenate all TRRP subsequent outputs and run the UPDT procedure only once.

#### EXECUTION BY SESSION

- Transaction Set code different from '\*\*\*\*\*'
- Target session: 'nnnnT' or '\*\*\*\*\*'

TRRP considers transactions that belong to the selected Transaction Set only.

- If you have selected a target session, transactions are generated for this session only.
- 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 (UY) once all transactions have been generated for a given session.

Generated transactions must then be used by the batch update procedure (UPDT) of the development database.

#### 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.

**NOTE:** 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).

#### RESULT

Transfer transactions formatted for the development database batch update procedure (UPDT).

## TRRP - User Input

User identification line (required)

Position	Length	Value	Meaning
2	1	'*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	5		Selection of target session(s):
		blank	. All target sessions (default), one session processed per TRRP execution. This value cannot be used when all Transaction sets are selected
		nnnnT	. Target session number (required)
		'*****'	. All target sessions processed in one TRRP execution
33	10		Selection of Transaction Set(s):
		llllllllll	Transaction Set code
		*****	All Transaction Sets
43	1		Formatting for UPDT
		'1'	Formatting
		' '	No formatting
44	1		Formatting for UPDP
		'1'	Formatting
		' '	No formatting

Sort Options line

Position	Length	Value	Meaning
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
7	1	' '	List by session
		'N'	No list by session

**TRRP - Description of Steps**  
INPUT RECOGNITION: PTU001

VSAM FILES CHECK: IDCAMS

PREPARATION OF EXTRACTION: PTUG60

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7JT	&INDUN..&BASE.TJ	Input	Sequential or compressed Journal
PAC7MB	&&TRRPMB	Input	User input
PAC7UY	&INDUV..SYTRPF.&USER	Input	Limited parameters
PAC7BM	&&PACXMB	Output	Extraction request for PACX
PAC7PJ	&&PACXPJ	Output	Temporary file
PAC7ET		Report	Statistics on transfer
PAC7DD		Report	Check on user

Return codes:

- 0: No error
- 8: Critical error (detailed in PAC7DD)

INITIALIZATION OF KSDS WORKING FILE: IDCAMS

EXTRACTION: PACX

This step extracts the transactions according to user inputs

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AY\$BASE	&INDUV..&BASE.AY	Input	Development Database extension data

Code	Physical name	Type	Label
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GU	&INDSV..BVPGU	Input	Administration Database user file
PAC7PJ	&&PACXPJ	Input	Transactions selected by the Journal
PAC7MB	&&PACXMB	Input	User input
PAC7BM	&&PACXBM	Input/Output	User input
SYSEXT	&&SYSEXT	Input/Output	Work file
PAC7MJ	&&PAC7MJ	Input/Output	Journal transactions (EXPJ)
PAC7WD	&&PACXWD	Input/Output	Extracted transactions
PAC7MV	&&MV	Output	Extracted transactions for UPDT
PAC7GY	&&GY	Output	Extracted transactions for UPDP
PAC7IA		Report	General program-stream printout
PAC7DD		Report	Printing of errors list on input transactions
PAC7EE		Report	Extractions report
PAC7EP		Report	Extractions report
PAC7EQ		Report	Extractions report
PAC7EZ		Report	Extractions report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK02		Sort	

Return codes:

- 0: No error
- 4: Error on user input (detailed in PAC7EE) or on the extractions for EXTR/EXUE (detailed in PAC7EZ)
- 8: Error on '\*' line (detailed in PAC7DD) or in EXLI (Database not available)

POSITIONING THE 'PROCESSING SESSION' ATTRIBUTE: PTUG61

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database data
PAC7MB	&&TRRPMB	Input	User input
PAC7UY	&INDUV..SYTRF.&USER	Input/Output	Limited parameters
PAC7ET		Report	Transfer statistics

**TRRP - Execution JCL**

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      PAC/TRANSFER -
/**      GENERATING THE TRANSFER TRANSACTIONS
/** -----
/**
/** ONCE THE TRANSFER FILE HAS BEEN BUILT, THE TRRP
/** PROCEDURE GENERATES TRANSFER TRANSACTIONS. THESE HAVE
/** THE SAME FORMAT AS BATCH UPDATE TRANSACTIONS
/** APPLICABLE BY THE UPDT PROCEDURE.
/**
/** INPUT :
/** - USER IDENTIFICATION LINE (REQUIRED)
/**   COL 2 : "*"
/**   COL 3 : USERIDXX
/**   COL 11 : PASSWORD
/**   COL 22 : (5 CAR.) SELECTION OF TARGET SESSION(S)
/**   COL 33 : (10 CAR.) SELECTION OF TRANSACTION SET(S)
/**   COL 43 : "1"  UPDT FORMAT
/**           " "  NO FORMAT
/**   COL 44 : "1"  UPDP FORMAT
/**           " "  NO FORMAT
/** - COMMAND LINE :
/**   COL 2 : "J"  LINE CODE
/**   COL 4 : " "  CHRONOLOGICAL LIST
/**           "N" NO CHRONOLOGICAL LIST
/**   COL 5 : " "  LIST BY USER
/**           "N" NO LIST BY USER
/**   COL 6 : " "  LIST BY LIBRARY
/**           "N" NO LIST BY LIBRARY
/**   COL 7 : " "  LIST BY SESSION
/**           "N" NO LIST BY SESSION
/** -----
/**
/**/BVPTRRP PROC BASE=$BASE,          CODE OF VAPAC DATABASE
/**   INDUV='$INDUV',                INDEX OF VSAM USER FILES
/**   INDSV='$INDSV',                INDEX OF VSAM SYSTEM FILES

```



```

//      INDSN='$INDSN',           INDEX OF NON-VSAM SYSTEM FILES
//      INDUN='$INDUN',         INDEX OF NON-VSAM USER FILES
//      OUT=$OUT,              OUTPUT CLASS
//*:   VSAMCAT='$VCAT',        VSAM USER CATALOG
//*:   SYSTCAT='$SCAT',       VSAM SYSTEM CATALOG
//      STEPLIB='$HLQ..SBVPMBR8', LOAD-MODULES LIBRARY
//      SORTLIB='$BIBT',      SORT LIBRARY
//      USER=,                USER CODE
//      CYL=5,                SORTWORK SPACE
//      UWK=$UWK,            WORK UNIT
//      SPAMV='(TRK,(30,10))', EXTRACTED TRANSACTIONS SPACE
//      SPAGY='(TRK,(30,10))', EXTRACTED TRANSACTIONS SPACE
//      SPAPJ='(TRK,(30,10))', SELECTED PJ FILE SPACE
//      SPAMB='(TRK,(5,1),RLSE)', SPACE OF COMMAND FILE
//      SPATD='(TRK,(50,10),RLSE)', SPACE OF CPNS FILE
//      PSBLIB='$PSBLIB',     LIBRARY OF PSB'S
//      DBDLIB='$DBDLIB',     LIBRARY OF DBD'S
//      RESLIB='$RESLIB',     IMS RESLIB
//      PROCLIB='$PRCLIB',    IMS PROCLIB
//      BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//      CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN,DCB=BLKSIZE=80
//PAC7MB  DD DSN=&&TRRPMB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY  DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PTUG60 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTUG60,PTUG60$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR

```

```

//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//         DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVP7GU,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7UY   DD DSN=&INDUV..SYTRPF.&USER,DISP=SHR
//PAC7JT   DD DSN=&INDUN..&BASE.TJ,DISP=SHR
//PAC7MB   DD DSN=&&TRRPMB,DISP=(OLD,PASS)
//PAC7BM   DD DSN=&&PACXMB,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=(TRK,(1,1)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
//PAC7PJ   DD DSN=&&PACXPJ,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=&SPAPJ,
//         DCB=(RECFM=FB,LRECL=170,BLKSIZE=17000)
//PAC7DD   DD SYSOUT=&OUT
//PAC7ET   DD SYSOUT=&OUT
//*-----
//MAXKEY   EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&&SYSEXT,DISP=(NEW,KEEP),
//         SPACE=(CYL,(3,3)),
//         LRECL=254,RECORG=KS,KEYOFF=0,KEYLEN=43
//MAXKEY   DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//*-----
//PACX     EXEC PGM=DFSRR00,REGION=$REGSIZ,
//         PARM=(DLI,BVPACX,PACX$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//         &IRLM),
//         COND=(0,NE,PTUG60)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR

```

```

//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT  DD SYSOUT=&OUT
//SYSOUX  DD SYSOUT=&OUT
//DDSNAP  DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//        DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//        BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//        BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON  DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE  DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN  DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR  DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU  DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7PJ  DD DSN=&&PACXPJ,DISP=SHR
//PAC7IA  DD SYSOUT=&OUT
//PAC7DD  DD SYSOUT=&OUT
//PAC7EE  DD SYSOUT=&OUT
//PAC7EP  DD SYSOUT=&OUT
//PAC7EQ  DD SYSOUT=&OUT
//PAC7EZ  DD SYSOUT=&OUT
//SYSEXT  DD DSN=&&SYSEXT,DISP=(OLD,DELETE,DELETE)
//PAC7MB  DD DSN=&&PACXMB,DISP=(OLD,PASS)
//PAC7BM  DD DSN=&&PACXBM,DISP=(,PASS),UNIT=&UWK,
//        DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7MM  DD DUMMY
//PAC7MJ  DD DSN=&&PAC7MJ,DISP=(,PASS),UNIT=&UWK,
//        SPACE=&SPAMV,
//        DCB=(RECFM=FB,LRECL=158,BLKSIZE=6320)
//PAC7TE  DD DUMMY
//PAC7RE  DD DUMMY
//PAC7RM  DD DUMMY
//PAC7WD  DD DSN=&&PACXWD,DISP=(,PASS),UNIT=&UWK,
//        SPACE=&SPATD,
//        DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SORTLIB DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(&CYL,1),,CONTIG)
//PAC7MV  DD DSN=&&MV,DISP=(,PASS),UNIT=&UWK,
//        SPACE=&SPAMV,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MR  DD DUMMY
//PAC7TD  DD DUMMY
//PAC7GY  DD DSN=&&GY,DISP=(,PASS),UNIT=&UWK,
//        SPACE=&SPAGY,
//        DCB=(RECFM=FB,LRECL=310,BLKSIZE=12400)
//PAC7UE  DD DUMMY

```

```

//*-----
//PTUG61 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTUG61,PTUG61$SUG,&BUF,
//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=((0,NE,PTUG60),(0,NE,PACX))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7UY DD DSN=&INDUV..SYTRPF.&USER,DISP=SHR
//PAC7MB DD DSN=&&TRRPMB,DISP=(OLD,DELETE)
//PAC7ET DD SYSOUT=&OUT
//*

```

## Update of the Development Database

The Development 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 development Database).

## Reinitialization of 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).







Part Number: DEPIM002301A - 6061

Printed in U.S.A.

(1P) P/N: DEPIM002301A - 6061

