VisualAge Pacbase 2.5

VA PAC 2.5 : UNIX
OPERATIONS MANUAL VOLUME II : ADMINISTRATOR'S GUIDE

DELIX002251A

Note

Before using this document, read the general information under "Notices" on the next page.

According to your license agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:
http://www.software.ibm.com/ad/vapacbase/support.htm

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

First Edition (July 1998)

This edition applies to the following licensed program:

- VisualAge Pacbase Version 2.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at:
http://www.software.ibm.com/ad/vapacbase/support.htm
or to the following postal address:
IBM Paris Laboratory
VisualAge Pacbase Support
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

NOTICES

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

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

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

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

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

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

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

TRADEMARKS

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

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

TABLE OF CONTENTS

1. OVERVIEW

THE ADMINISTRATOR'S GUIDE: OVERVIEW

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

These procedures relate mainly to the following operations fields:

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

PRESENTATION OF PROCEDURES

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

The presentation of a procedure contains the following:

. General introduction, including

  -presentation

  -execution condition(s)

  -actions to be taken in case of abnormal execution

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

. Execution JCL.

1.1. USER IDENTIFICATION (*)

USER IDENTIFICATION '*' LINE

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

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

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

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

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

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

1.2. ACCESS RIGHTS: BATCH-PROCEDURE AUTHOR. OPTION

'BATCH-PROCEDURE ACCESS AUTHORIZATION' OPTION

PRINCIPLE OF THE OPTION

This option is used to grant each user the access.

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

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

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

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

CONSEQUENCE

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

OPTION ACTIVATION

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

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

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

BATCH PROCEDURE ACCESS AUTHORIZATION TABLE

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

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

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

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

The following procedures do not require an authorization access check:

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

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

1.3. ABNORMAL ENDINGS

ABNORMAL EXECUTION ENDINGS

A BATCH program execution may abend.

For example, input-output errors on the system files or on the database files cause the interruption of the current program and the display of the following messages:

```
PROGR : pppppp   INPUT-OUTPUT ERROR : FILE ff   OP : oo
STATUS : nn
```

In most cases, examining the status and type of operation allows you to find the cause of the abend.

The tables below indicate standard values for the status and type of operation.

NN	STATUS	OO	OPERATION
21	SEQUENCE ERROR		
22	DUPLICATE KEY	W	WRITE
23	NO RECORD FOUND	RW	REWRITE
24	BOUNDARY VIOLATION	RU	READ UP
30	SYSTEM ERROR	OP	OPEN
34	BOUNDARY VIOLATION (SEQ.)	CL	CLOSE
35	FILE NOT FOUND	D	DELETE
92	LOGIC ERROR (FOR EX. OPEN AN ALREADY OPENED FILE)	R	READ
		P	START
93	LOCKED FILE	RN	READ NEXT
95	INVALID OR INCOMPLETE FILE DEFINITION		

Some errors, other than input-output errors on a Database file, may also cause the following message to be displayed:

Run Time Error nnn (Where 'nnn' is the error number.)

Run Time Error 013 is the most common error. It means that the procedure did not find an input file. In order to find out which file is missing, enter the SET command. This will display the list of allocated files. You can also consult the procedure description in the corresponding Chapter of this Manual. Then, compare this list with the contents of the directories involved.

Most often, it is the Input Transactions file that is missing (in the "release"\INPUT\"db_name" directory : MBxxxx where xxxx is the procedure specific code).

The following subchapter contains the list of the most frequent errors. Each Run Time Error is accompanied by a short explanatory message.

If a Run Time Error does not appear in the following list, or if the message is insufficient and the type of error signals a direct problem in the system programs, contact your Technical Support (Help Desk) and save all listings that could help in analyzing the problem.

1.4. STRUCTURE OF PROCEDURE COMMAND FILES

STRUCTURE OF THE PROCEDURES COMMAND FILES

The BATCH procedures command files are created in the $PACDIR/batch/proc directory at installation time.

PROCEDURE ADAPTATION TO SITE CONSTRAINTS

The VisualAge Pacbase Database manager sometimes has to modify the batch procedures command files.

For example, if he/she wants to seperate the AN and AR files onto two different disks or move the AE file, the resulting modifications in the command files can be considerable.

This is why the procedures (batch or monitor start-up) are there to ease all modifications of the standard installation and to minimize the procedure adaptations linked to the operation constraints.

The objective of this subchapter is to analyze a batch procedure in order to explain how it works and to guide the user in its posible adaptations.

STRUCTURE OF A BATCH PROCEDURE

1. The parameters

- General case:

The BATCH procedures only use one parameter:

```
the database name.
```

- Multi-users procedures case:

Two extra parameters were added to the procedures likely to be executed simultaneously by several users: (ACTI, DCOB, ECSP, EMLD, EMUP, PACX, GETA, GETD, GETI, GPRT, PPAF, PQCA, PQCE, PRPE, RVDE, RVKE, SADM, TRUV, UPDT, XPAF, XPDM, YSMC) :

The first of these two parameters is used as a root of the files of the 'input' and 'tmp' directories, to differentiate them according to the user.

The second parameter is used as a suffix or as a sub-directory of the 'input' and 'tmp' directories.

These parameters are valorized or not at the start-up of each BATCH procedure:

. if the two parameters are blank, the assignment of 'input' and 'tmp' directories' files is not changed,

. the first parameter can be assigned a value while the second is left blank,

. the second parameter cannot be assigned a value if the first is blank,

. the first parameter must contain a maximum of two characters so that the files are visible in DOS or OS/2,

. if the second parameter begins with the character '/', it represents a sub-directory of 'input' and 'tmp', if not it is a suffix of the 'input' and 'tmp' directories; in all cases the creation of the resulting directories is the user's responsibility.

These two parameters are used in the PACINPUT.ini and PACTMP.ini commands files, called in each procedure, to initialize the PACINPUT and PACTMP environment variables (cf paragraph "FILE ASSIGNMENT AND CODING").

2. Display and verification of parameters

The execution of a procedure begun by the execution of the USAGE.ini commands file:

```
. $PACDIR/batch/proc/USAGE.ini
```

This file is created at installation in the PACDIR/batch/proc directory.

The USAGE.ini commands file controls the parameters along with the procedure and possibly positions the PACRAD and PACSUF environment variables.

These two variables are used in the PACTMP.ini and PACINPUT.ini commands files and corrspond to root parameters and suffixes of the multi-users procedures.

If an abend is detected, USAGE.ini displays a corresponding error message and stops the procedure with a 20 return code.

If not the execution of the procedure continues by displaying the directories' assignments.

In order to visualize this display, at least during an installation test, the execution must be suspended by a call to the commands file:

```
sh $PACDIR/batch/proc/MSGPAUSE.ini
```

The MSGPAUSE.ini file, created during installation in the $PACDIR/batch/proc directory contains:

```
echo **********  Verify your parameters  *********
echo Press Control_C to stop execution
echo Press Return to continue
read REPAUSE
```

If you do not want to suspend execution, you must modify the contents MSGPAUSE.ini deleting the line: "read REPAUSE".

4. Assignment and coding of files

Each step must be assigned adequate files.

- THE DATABASE FILES

These assignments are carried out via the commands files' call, created at installation in the directory:

```
$PACDIR/assign/"database_name".
```

Example of the assignment of the AE file:

```
. $PACDIR/assign/$1/PAC7AE.ini
```

The main interest in these files is to centralize the assignment of each database file in a single place.

The user who wants to modify the standard location of a file only has to adapt the assignment file.

Note: the same files are used when starting up the servers.

- THE BACK-UP FILES

As for the database files, these assignments are carried out via the commands files' call, created at installation in the directory:

```
$PACDIR/assign/"database_name".
```

Example of the assignment of the PC file:

```
. $PACDIR/assign/$1/PACSAVPC.ini
```

By default, the PE back-up (user parameters) is located in $PACDIR/save and the others (PC, PJ, PG and PP) in $PACDIR/save/"database_name".

All of the batch procedures which use one of the back-up files are standardized as the files name:

```
input back-up (consulted) = Px

output back-up (created by the procedure) = Px.NEW
(with x = C, E, J, G ou P)
```

This simplifies the management of these files (see for example the paragraph "Back-up files management" a little further on).

- THE TRANSACTION FILES

All of the transaction files used for procedure input are coded MBxxxx (xxxx being the procedure name).

All the transaction files created as procedure output are coded MVxxxx (xxxx being the procedure name). It concerns, for example, the transactions generated by the extraction procedures.

The location of transaction files is determined by the PACINPUT environment variables, positioned in each procedure by the PACINPUT.ini command file call:

```
. $PACDIR/assign/$1/PACINPUT.ini
```

The PACINPUT.ini file is created when VisualAge Pacbase is installed and when a database is created in the directory:

```
$PACDIR/assign/'database_name".
```

It contains:

```
# Assignment script of the PACINPUT environment variable
# ('input' directory)

# Parameters description:    : $0      = procedure name
#                              $1      = database name
#                              $PACRAD = file root
#                              $PACSUF = directory suffix

PACINPUT=$PACDIR/input$PACSUF/$1/$PACRAD
export PACINPUT
```

Example of assignment in the EXTR procedure:

```
PAC7MB=$PACINPUT'MBEXTR'
export $PAC7MB

PAC7MV=$PACINPUT'MVEXTR'
export $PAC7MV
```

- THE OUTPUT REPORTS

All of the procedures output reports are created in the temporary files directory and their names start with their respective procedure codes.

This means that they can be easily consulted or printed (print SAVE*.* par example).

More precisely, the reports are coded on six characters plus an extension, in the following manner:

. the first four characters correspond to the procedure code (SAVE in PROCSAVE),

. the next two correpsond to the last two characters of the file (EU in PAC7EU),

. the extension represents the last three characters of the program code (500 in PTU500).

```
Example: SAVE procedure, PTU500 program            report PAC7EU
--> SAVEEU.500
            report PAC7DS     --> SAVEDS.500
```

- THE TEMPORARY FILES

See the subchapter "ADVICE ON USE".

5. End of the procedure without error

If no error is detected, the message "End of procedure" is displayed.

6. End of procedure with error

As soon as an error is detected in a step, the following steps are not executed. The name of the program with the error is displayed and if possible the type of error.

The procedure executes the ERRPAUSE.ini file which allows you to stop the procedure and visualize the error.

The ERRPAUSE.ini file created at installation contains:

```
    echo "Press the Return key to continue"
    read REPAUSE
```

7. Back-up files management

All the files creating one of the back-ups call a commands file at the end of a procedure without error.

These files are in the $PACDIR/assign/"database_name" directory (they include that which manages the PE backup) and are called PxBACKUP.ini (x = C, E, J, G or P).

They are created when the database is created and contain (PJBACKUP.ini for example):

```
# Rotation script of the journal back-up file
. $PACDIR/assign/d400/PACSAVPJ.ini
if   -f "$PACSAVPJ"
then
   mv -f $PACSAVPJ $PACSAVPJ'-1'
fi
mv -f $PACSAVPJNEW $PACSAVPJ
```

Characteristics of the PxBACKUP files:

. proceed by 'mv' to avoid copies which can be very long,

. ensure a rotation on the last two versions of the back-up files,

. guarantee that the Px file is definitely the last back-up (Px being systematically used as procedure input),

These files do not claim to cover all the operation constraints of all sites. The database manager generally has to adapt them, taking the characteristics above into account.

Use of the PxBACKUP files:

```
. PCBACKUP.ini : used in the SAVE, MLIB,
                 REOR and QREO procedures.
. PEBACKUP.ini : used in the PARM procedure.
. PJBACKUP.ini : used in the ARCH procedure.
. PGBACKUP.ini : used in the SVAG procedure.
. PPBACKUP.ini : used in the SVPE procedure.
```

1.5. RECOMMENDATIONS

ADVICE ON USE

The objective of this subchpater is to make the person responsible for the database aware of the specifics of the VisualAge Pacbase procedures executed on the UNIX system.

Temporary files

For each procedure the user should consult the corresponding chapter for a detailed description of these files.

In all cases, enough disk space should be freed in the chosen user directory to ensure that the procedure runs as normal.

Temporary sort files

When a program executes a sort, the COBOL routines called also use a temporary file independent of those listed above.

This file is created by default in the /usr/tmp directory.

Its size can be 3 or 4 times the size of the file to be sorted.

If the default directory is too small, the TMPDIR directory assigns another directory for the temporary sort files:

```
TMPDIR=/tmp2
export TMPDIR
```

Location of the temporary files

The location of the temporary files is determined by the PACTMP.ini command file call:

```
. $PACDIR/assign/$1/PACTMP.ini
```

The PACTMP.ini file is created when VisualAge Pacbase is installed and when a database is created in the directory:

```
$PACDIR/assign/'database_name"
```

It contains:

Assignment script of the temporary files

```
  Parameters' description    : $0      = procedure name
                               $1      = database name
                               $PACRAD = file root
                               $PACSUF = directory suffix

  Directory of the 'tmp' temporary files
PACTMP=$PACDIR/tmp$PACSUF/$1/$PACRAD
export PACTMP

  Directory of the temporary sort files
  (this directory must be 3 times the size of the file to be
   sorted)
TMPDIR=$PACDIR/tmp/$1
export TMPDIR

  Type of sort used for REOR and QREO
  (PACSORT=cobol -> cobol sort,PACSORT=unix -> unix sort)
PACSORT=cobol
export PACSORT
```

The PACTMP.ini file initializes the PACTMP environment variable to assign the VisualAge Pacbase temporary files.

Example of assignment in the EXTR procedure:

```
PAC7EE=$PACTMP'EXTREE.S10'
export $PAC7EE
```

The PACTMP.ini file initializes the TMPDIR environment variable to assign the temporary sort files directory.

The PACTMP.ini also initializes the PACSORT environment variable, which allows you to choose the type of sort used (only for REOR and QREO procedures):

. either a COBOL Micro Focus sort (default choice), which involves the TMPDIR directory which must be at least three times the size of the file to sort and which limits the size of this file to 600 Mb due to the constraints of the UNIX system.

. or a UNIX sort which is quicker but uses more CPU time.

This type of sort involves the TMPDIR directory which must be at least twice the size of the file to sort and which can sort files up to 2 Gb.

NOTE: GENERAL COMMENTS

1. Each procedure must be passed parameters. All the parameters which may be called in a procedure must be present, even if they are not actually used.

2. When user input is expected in a procedure, even if it is optional, the corresponding transaction file must be present when the procedure is being executed.

3. No protection is guaranteed in the case where a BATCH procedure updating the database system or evolving files is started up when users are interactively updating these same files. One person (the database manager) must be able to start-up the batch procedures updating the database. He/she therefore must ensure the database data (closing the on-line servers for example).

4. The temporary workfiles created by the batch procedures are automatically destroyed at the end of the procedure, except if there was an abend and return code different than 1.

1.6. SUBMISSION OF PROCEDURES

BATCH PROCEDURES START-UP

The batch procedures must be submitted from a UNIX machine.

PROCEDURES ERROR MANAGEMENT

If an error is detected in a procedure, the procedure is stopped with a return code other than zero. This code is recoverable in the $? variable directly after the procedure start-up command.

This prevents the next procedure from being executed if there is a string of procedures.

1.7. LIST OF RUN-TIME ERRORS

LIST OF RUN-TIME ERRORS

This list is a reminder of the most common errors and their meaning.

Number	Meaning
004	Invalid file name
005	Invalid device specification
007	No more disk space
009	Directory full or does not exist
013	File not found
026	Block I-O error
027	Device not available
028	Disk space exhausted
033	Physical I-O error
105	Memory allocation error
116	Cannot allocate memory
135	File not found
150	Program abandoned on user request
157	Not enough program memory: object file too big to load
170	System program not found
173	Called program file not found
188	File name too long
198	Not enough program memory: object file too large to load
207	Machine does not exist on the network
208	Network communication error
209	Network communication error
221 !	
222 !>	Error during a SORT
223 !	

2. MONITOR START-UP

2.1. TP: ON-LINE SERVER START-UP

ON-LINE MONITOR START-UP

The on-line monitor, whose executables (pactp, pacserver, paclaunch) are in the $PACDIR/bin directory allows the following:

- to set the on-line server(s) in active or inactive mode,
- to supply information concerning the on-line server(s),
- to purge workstation attached to an on-line server,
- to purge the on-line server(s).

To carry out the operation listed above the on-line monitor commands interpretor (pactp) is started up.

There are two posssible modes:

- the 'command' mode,
- the 'shell' mode.

For the 'command' mode type:

```
pactp <command>
```

To return to 'shell' mode type:

```
pactp -s
```

The 'command' mode is used to insert the on-line monitor commands in a commands file.

For example, the com_paclan file contains the following command:

```
# display the servers' status
pactp info
# start the tp1 on-line server
pactp start tp1
# display information on the tp1 server in ten seconds
pactp info tp1
sleep 10
# stop the tp2 on-line server
pactp shutdown tp2
# start the 003 station purge command dependant on the tp1
server
pactp purge tp1 003
```

The advantage of the 'shell' mode is that you avoid re-starting the interpretor for each command and so you avoid delays between commands.

The commands available are:

```
-debug      (activation/deactivation of degugger mode)

-exit       (exit from 'shell' mode),

-help       (help on a command's syntax),

-info       (information on the server(s)),

-purge      (purge the workstation),

-purge_server (purge the on-line server),

-shutdown   (stop the server without confirmation),

-start      (start the server),

-stop       (stop the server with confirmation),

-trace      (activation/deactivation of the trace),

-view       (display the server's status).
```

A detailed description of the commands is given below in the paragraph 'Description of the commands'.

EXECUTION CONDITIONS

The PACDIR environment variable must be initialized (see chapter "INSTALLATION" in the "VisualAge Pacbase - UNIX - ENVIRONMENT & INSTALLATION" manual.)

The COBPATH environment variable must contain the access path for on-line modules: "$PACDIR/tp/gnt".

ON-LINE SERVER CONFIGURATION

When started each server executes "server_name".ini-type command file to assign any environment file it needs.

This file is created when the server is created in the $PACDIR/assign/tp_server directory.

. On-line server environment variables:

```
 - BASENAME          : database name,
 - PAC7LB and PAC7BD: BATCH communication files,
 - STATIONS          : maximum number of workstations,
 - PACPAUSE          : workstation interrogation time in
                       milli-seconds,
 - PACSOCKET         : socket number,
```

```
. Database environment variables:

 - PAC7AN            : index file
 - PAC7AR            : data file
 - PAC7AE            : error messages file
 - PAC7AG            : print-generation file
 - PAC7AJ            : journal file
 - PAC7DC            : DSMS file
 - PAC7HE            : layout workfile
 - PAC7ZS            : database workfile
 - SEMLOCK           : serialization lock of concurrent
                       updates.
```

NOTE:

When creating or deleting on-line servers, it is strongly advised to use the pacadmin procedure.

DESCRIPTION OF THE COMMANDS

Debug command

This command activates or deactivates the 'debug' mode. The server name must be passed as a parameter, followed by "on" or "off" to activate or deactivate the 'debug' mode.

A results file is updated. It is called:

```
"server_name_process_number".SPY
```

and is in the $PACDIR/log directory.

EXAMPLE: start the debug mode on an on-line server called tp1

```
pactp debug tp1 on       (in command mode)
PACBASE : debug tp1 on  (in shell mode)
```

EXAMPLE: stop the debug mode on an on-line server tp1

```
pactp debug tp1 off      (in command mode)
PACBASE : debug tp1 off (in shell mode)
```

Exit command

This command is used to exit the shell mode (commands interpretor) previously done by pactp -s.

Help command

This command displays help concering an on-line server administation command. If it is not given a parameter, the list of available commands is displayed. If a command is specified, the help for this command is displayed on the screen.

EXAMPLE: request help on the syntax of the start command

```
pactp help start      (in command mode)
PACBASE : help start (in shell mode)
```

Info command

This command displays information on the on-line server(s).

This command followed by a server name (info "server_name") gives information from the server's configuration files:

```
        $PACDIR/assign/on-line_server/"server_name".ini
and     $PACDIR/assign/"database_name"/"PAC7xx".ini
```

This command alone can give the following information:

- active (if the server is active),
- inactive (is the server is inactive),
- not configured (if the server is declared at the $PACDIR/assign/on-line_server/paclanx.srs level but does not have the configuration file situated at the $PACDIR/assign/on-line_server/"server_name".ini) level,
- Error (if the server is stopped abnormally).

Purge command

This command purges a workstation (i.e. it deconnects a workstation).
If the interpreter (pactp) is in command mode, the syntax for the purge command is as follows:

"pactp purge <server_name> <no. of the station to purge>".

In shell mode there are two syntaxes for the purge command:

- "purge <no. of the station to purge>" if the prompt is different from "PACBASE:" (the prompt taking the value of server name),
- "purge <server_name> <no. of the station to purge>" if the prompt is "PACBASE:".

Purge server command

This command purges an on-line server in the case of an abend. The syntax for the purge_server command is as follows:

"pactp purge_server <server_name>".

In shell mode the syntax for the command is:

"purge_server <server_name>".

N.B.: This command deletes the communication files in the $PACDIR/commun and $PACDIR/tp/save directories and the messages (tables IPC) associated with the server.

Shutdown command

This command stops the on-line server. The server's name must be passed as a parameter of the command. A conversations backup file is created in the $PACDIR/tp/save directory. The name of this file is that of the on-line server with a .ts extension.

EXAMPLE: stop the tp1 on-line server

```
pactp shutdown tp1       (in command mode)
PACBASE: shutdown tp1  (in shell mode)
```

Start command

This command starts the on-line server. The server's name must be paseed as a parameter of the start command. The initialization file corresponding to the server (on-line server name with .ini extensiion) must be in the $PACDIR/assign/tp_server directory. The server start-up program, paclaunch, starts the pacserver process. If problems arise (on-line server blockage), the pacserver process can always be killed using the kill -15 command or kill -9 followed by deleting the server messages with the ipcs -q and ipcrm -q commands.

EXAMPLE: starting the tp1 on-line server

```
pactp start tp1       (in command mode)
PACBASE: start tp1  (in shell mode)
```

Stop command

This command stops an on-line server. The server's name must be passed as a parameter of the stop command. A conversations backup file is created in the $PACDIR/tp/save directory, whose name is that of the on-line server with a .ts extension. You have to confirm stopping the server.

EXAMPLE: stop the tp1 on-line server
PACBASE: trace tp1 off (in shell mode)

View command

This command displays information concerning an on-line server: the list of connected workstations, the user code, the name of the on-line program executed or being executed, the time of this program start-up and its execution time in milliseconde.

EXAMPLE: display information about the tp1 on-line server

```
pactp view tp1       (in command mode)
PACBASE: view tp1  (in shell mode)
```

2.2. PACPARM: PARM-PEI TRANSACTION START-UP

STARTING UP THE PARM TRANSACTION

The "database name" pacparm command starts up the PARM transaction on the "database name" database.

PARM, reserved to the database administrator, can be used on the UNIX machine.

The monitor program, pacparm, is in the $PACDIR/bin directory.

The environment variables configuration file, pacparm.ini, and the screen configuration file, pacparm, are both in the $PACDIR/assign/monitors directory.

The other configuration files (PAC7xx.ini) are in the $PACDIR/assign/"database name" directory.

END OF CONVERSATION

Exit the PARM-PEI transaction: when the user exits the transaction (F6 or CH: FT), the initial grill is displayed; to delete it, press the F12 function key, or an equivalent (see subchapter "Workstation configuration").

ACTIVATING DSMS CONTROL

Users with DSMS who wish to activate the control of their VisualAge Pacbase database must indicate this on the PARM transaction PK screen.

In the case of several VisualAge Pacbase databases which must be controlled by different DSMS databases, the correspondance is made by the DC file assignation.

Refer to the DSMS Operations Manual, chapter "INSTALLATION", subchapter "Putting a VisualAge Pacbase Database under DSMS Control" for more information.

EXECUTION CONDITIONS

The PACDIR environment variable must be initialized (refer to the "VisualAge Pacbase - UNIX : Environment and Installation" manual, chapter "Installation".

The COBPATH environment variable must contain the access path to the on-line module "$PACDIR/tp/gnt".

MONITOR CONFIGURATION

When it is started the monitor executes the pacparm.ini commands file to assign the environment variables it will need.

This file is created when the server is created in the $PACDIR/assign/monitors directory.

. PARM monitor environment variables:

```
 - RADICAL          : COBOL programs prefix (D8)
                      and language code
                      (F = French, A = English)
 - FPARAM           : screen configuration file
```

```
. Database environment variables:

 - PAC7AN           : index file
 - PAC7AR           : data file
 - PAC7AE           : error messages file
 - PAC7AB           : batch production environ. file
 - PAC7AC           : on-line production environ. file
 - PAC7DC           : DSMS file
 - PACHELP          : backup file
 - SAVESCR          : backup file
```

WORKSTATION CONFIGURATION

The configuration file, pacparam, which is in the $PACDIR/assign/monitors directory, contains the screen and keyboard parameter descriptions.

This file comprises three parts:

- a list of codes for the display configuration (screen attributes).
- a list of parameters describing the behaviour of the character input function.
- a correspondance table between the workstation controls and the PARM functions.

Display parameters

There are eleven display parameters: two color parameters (characters' color and background color) and nine parameters for managing the screen attributes.

The two colors available on the monochrome workstation are black (B) and white (W). The first color is for the characters, the second is the background color.

There are four possible values for the attributes:

. normal mode	N
. semi-intensity mode	D
. double intensity or brilliant mode	B
. reverse mode	R

The nine screen attributes are for the following fields:

```
. attribute for normal and protected fields
. attribute for protected and brilliant fields
. attribute for normal input fields
. attribute for brilliant input fields
. attribute for current field
. attribute for normal highlighted fields
. attribute for brilliant highlighted fields
. attribute for normal flashing fields
. attribute for brilliant flashing fields
```

Input function parameters

There are five parameters for the input function, which can have the values N or Y:

. the first is not used in this release.

. the second, if Y, indicates that the cursor will move to the next field when the current field is filled.

. the third, if Y, allows you to reposition the cursor on the first field of the screen after the last field.

. the fourth, if Y, allows you to delete the characters at the end of the current field in insertion mode.

. the last is not used in this release.

Keys correspondance table

The last part of the pacparm file describes the correspondances between certain keyboard key sequences and the functions of the PARM monitor.

In the current release only the control keys are available; these are:

```
          ctrl_A, ctrl_B, ..., ctrl_Z.

The PARM functions available are:

     . Enter        when the Enter key is pressed
     . Clear        clear the screen
     . PF1..PF24    function keys 1 to 24
     . Tab          move to the next input field
     . BackTab      move to the previous input field
     . NL           new line
     . Home         move to the first input field
     . End          move to the last input field
     . BeginF       move cursor to the start of the field
     . EndF         move cursor to the end of the field
     . Curs-U       move cursor up
     . Curs-D       move cursor down
     . Curs-L       move cursor to the left
     . Curs-R       move cursor to the right
```

```
. BSpace       delete the character before the cursor
. Insert       enter/exit insertion mode
. Delete       delete the character under the cursor
. DelEOF       delete to the end of the current field
. DelINP       delete the data of the current field
. Recover      re-display the screen
```

By default, certain controls are already assigned in the configuration files:

```
ctrl_H      BSpace
ctrl_I      Tab
ctrl_J      NL
ctrl_M      Enter
```

Other controls, like ctrl-C or ctrl-Z must be used with care as they can come into conflict with the workstation configuration, modifiable by the stty command.

N.B.: The programming of the function keys is linked to the termingo configuration file.

Therefore it is advisable to refer to this file when programming the function keys.

2.3. BATCH SERVER START-UP

BATCH MONITOR START-UP

The BATCH monitor, whose executables (pacbatch, pacbatsrv) are in the $PACDIR/bin directory, allow for

- activating or deactivating the BATCH server(s),
- supplying information concerning the BATCH server(s),
- purging BATCH server(s).

To carry out the operations listed above the BATCH monitor commands interpreter (pacbatch) is executed

The commands interpreter can be started up in two modes:

- command mode,
- shell mode.

For command mode type:

```
pacbatch <command>
```

To change to shell mode type:

```
pacbatch -s.
```

Command mode inserts the BATCH monitor commands in a commands file.

For example, the com_paclan file contains the following commands:

```
# display the servers state
  pacbatch info
# start-up the bat1 BATCH server
  pacbatch start bat1
# display information on the bat1 server for ten seconds
  pacbatch info bat1
  sleep 10
# stop the bat2 BATCH server
  pacbatch stop bat2
```

The advantage of shell mode is that it avoids having to re-start the interpreter for each command and so avoids any delay between commands.

The commands available are:

```
-exit      (exit shell mode),

-help      (help on a command's syntax),

-info      (information on the server(s)),

-purge_server (purge the BATCH server),

-shutdown  (stop the BATCH server without confirmation),

-start     (start the server),

-stop      (stop the BATCH server with confirmation),

-view      (display the server status).
```

There is a detailed description of the commands listed above in the paragraph 'Description of the commands'.

The BATCH functions executed by the server are the same as those for the GPRT procedure (.gnt files in the $PACDIR/batch/gnt directory).

To process the print-generation requests on a database, it is possible to start-up several batch servers which share the same couple of files LB and BD and therefore request processing.

In the case of several databases, one or more batch servers must be defined for each database.

EXECUTION CONDITIONS

The PACDIR environment variable must be initialized (refer to chapter "INSTALLATION" in the "VisualAge Pacbase - UNIX: Environment & Installation" manual).

The COBPATH environment variable must contain the access path to the BACTH functions "PACDIR/batch/gnt".

REQUEST RESULTS

When a workstation submits a print-generation from the GP screen, the resulting screen displays the following:

```
JOB STREAM BUILT - NUMBER : nnnnn
```

The files created by the CPRT procedure are inserted in the "user code" sub-

directory of the "users" directory.
Their code is made up of the request number followed by the type of file generated (see the description of the GPRT procedure for more details on this coding).

```
Example: in $PACDIR/users/jean, we find:       00055.ia
                                               00055.gp
                                               00055.in
                                               etc...
```

Only the "useful" files are kept: report on the execution of the print-generation string (.ia), file of entities (.id) and generated entities (.gp, .ge, .im, ...).

Files whose suffix begins with "x" (.xgi, .xgm and .xgn) are temporary files, which the user may want to recover for specific processes. The are not automatically deleted at the end of a job.

RESULTS FILE PROCESSING

The PLBTAGP variable declared in the server configuration file allows a call to the PACAGP commands file after the print generations.
The call to this file allows certain files, whose nature varies according to the environment, to be automated.

An example of the PACAGP file is supplied at installation in the $PACDIR/batch/proc directory. This file must be modified to take the specifics of each VisualAge Pacbase site into account.

In the example given, the comments explain in particular the parameters that the batch server passes to this commands file and which are therefore usable for processing the results files.

By default, the PACAGP file is in the $PACDIR/batch/proc directory.

GENERATED COBOL SOURCE FILES BREAK-UP

When several generations of the same type are carried out in a single job, all the sources are produced in the same output file.

The pacsplit program separates the COBOL code generated into distinct files.

This program is described in the GPRT chapter, subchapter "Interface with Micro Focus Workbench".

INTERFACE WITH GDT-PC

For the processing of generation results files by the GDT-PC tool, refer to the GPRT chapter, subchapter "GDT-PC Interface".

INTERFACE WITH MICRO FOCUS WORKBENCH

For the processing of generation results files with compilation by the Micro Focus Workbench, refer to the GPRT chapter, subchapter "Interface with Micro Focus Workbench".

BATCH SERVER CONFIGURATION

When started up each server executes a "server_name".ini- type commands file to assign the environment variables it needs.

This file, created when the server is created in the PACDIR/assign/batch_server directory.

```
. BATCH server environment variables:

 - BASENAME          : database name
 - PAC7LB and PAC7BD: BATCH communication files
 - PLBTPAR           : file with list of assignments made by
                       the BATCH server
 - SYSOUT            : output files sub-directory
 - PLBTDEL           : temporary files delete option
                       (YES = delete, NO = keep)

 - PLBTLG            : language option for printing
                       print-generation reports
                       (F = French, A = English)
 - PLBTMON           : string monitor code of the
                       print-generation string
                       (PACB = French, PACBE = English)
 - PLBTAGP           : PACAGP file execution option
                       (YES = PACAGP call, NO = no call)

. Database files environment variables:

 - PAC7AE            : error messages file
 - PAC7AN            : index file
 - PAC7AR            : data file
 - PAC7AG            : print-generation commands file
 - PAC7AP            : user parameters file
 - PAC7AB            : batch production environment file
 - PAC7DC            : VisualAge Pacbase elements DSMS file
 - PAC7GS            : extraction schemas file
 - PAC7AC            : on-line production environment file
 - PAC7SC            : batch skeleton file
 - PAC7SG            : on-line skeleton file
 - PAC7SS            : client/server skeleton file
 - PAC7SP            : XPAF skeleton file
 - PAC7SF            : XPAF fixed skeleton file
 - PAC7SR            : REVERSE skeleton file
 - PAC7LG            : OLSD error messages file
 - PAC7LK            : PACBENCH C/S error messages file
```

NOTE:

It is strongly advised to use the pacadmin procedure when creating or deleting batch servers.

DESCRIPTION OF THE COMMANDS

Exit command

This command is used to exit shell mode (commands interpreter) previously done by the pacbatch -s command.

Help command

This command displays help concerning a BATCH server administration command. If it is not passed a parameter, the list of available commands is displayed. If a particular command is specified, the help for this command is displayed.

```
EXAMPLE: request to display the syntax of a command         start

         pacbatch help start     (in command mode)
         PACBASE : help start    (in shell mode)
```

Info command

This command displays information on the BATCH server(s).

This command followed by the server name(info "server_name") gives information from the server configuration files, which are in:

```
         $PACDIR/assign/batch_server/"server_name".ini
and      $PACDIR/assign/"database_name"/"xxxxxx".ini
```

This command alone (info) can give the following information:

- active (if the server is active),
- inactive (if the server is inactive),
- not configured (if the server is declared at the $PACDIR/assign/batch_server/paclanx.srs level but does not have the configuration file at the $PACDIR/assign/batch_server/"server_name".ini level),
- Error (if the server is stopped abnormally).

Purge_server command

This command purges a BATCH server in the case of an abend. The syntax of the purge_server command is as follows:

"pacbatch purge_server <server_name>".

In shell mode the syntax of the command is:

"purge_server <server_name>".

N.B.: This command deletes the messages (tables IPC) associated with the server.

Shutdown command

This command stops the BATCH server without requesting confirmation. The server's name must be passed as a parameter of the command.

EXAMPLE: stop the bat1 BATCH server

```
pacbatch shutdown bat1   (in command mode)
PACBASE: shutdown bat1   (in shell mode)
```

Start command

This command starts up a BATCH server. The server's name must be passed as a parameter of the command.
The initialization file corresponding to the server (BATCH server's name with .ini extension) must be in the $PACDIR/assign/batch_server directory.
The server start-up program, pacbatch, starts up the pacbatsrv process itself.
In case of problems (crash of the BATCH server), the pacbatsrv process can always be killed using kill -15, or kill -9 followed by deleting the server messages using the ipcs -q and ipcrm -q commands.

EXAMPLE: starting up the bat1 BATCH server

```
pacbatch start bat1    (in command mode)
PACBASE: start bat1    (in shell mode)
```

Stop command

This command stops a BATCH server. The server's name must be passed as a parameter of the stop command.

EXAMPLE: stop the bat1 BATCH server

```
pacbatch stop bat1   (in command mode)
PACBASE: stop bat1   (in shell mode)
```

View command

The view option displays the state of a given BATCH server. For each job submitted to the server, the following information is displayed:

- line number,
- job number,
- user code,
- workstation number,
- date and time of job submission,
- processing start time,
- processing end time,

EXAMPLE: display information on bat1 BATCH server

```
pacbatch view bat1       (in command mode)
PACBASE: view bat1       (in shell mode)
```

2.4. PACWST: UNIX WORKSTATION START-UP

START-UP OF A UNIX 'DUMB' WORKSTATION (PACWST - PACWSTSOCK)

COMMUNICATION VIA FILES

In the case of communication via communication files, dialog between the workstation user and the "server_name" on-line server is carried out by using the command:

```
pacwst <-trace> "server_name"
```

COMMUNICATION VIA SOCKET

In the case of communication via the interface socket, the start-up of dialog between the workstation user and the on-line server is carried out by using the command:

```
pacwstsock <-trace>
```

The on-line server to which the station is connected is determined by the station's environment variables:

"PAC_HOST_SERVER_NAME", which contains the name of the UNIX machine which starts-up the server.

"PAC_HOST_SERVER_SOCKET", which contains the number of the communication socket used by the server.

In other words, the station will connect to the server using the "PAC_HOST_SERVER_SOCKET" communication socket on the machine called "PAC_HOST_SERVER_NAME".

The "pacwst" and "pacwstsock" commands are in the $PACDIR/bin directory.

"TRACE" OPTION

The trace option activates the trace.

The journal file supplied by this option is in the form:

```
wstyymmdd.LOG (yy = year, mm = month, dd = day)
```

It is in the $PACDIR/log directory.

END OF CONVERSATION

Exit the transaction: when the user exits the transaction (F6 or CH: FT), the initial grill is displayed.

To delete it press the F12 key, or the equivalent (refer to subchapter START-UP OF THE PARM-PEI TRANSACTION, paragraph WORKSTATION CONFIGURATION).

ENVIRONMENT

The "pacwst" environment variable configuration file is called "pacwst.ini", that of "pacwstsock" is called "pacwstsock.ini".

These two files are created at installation in the PACDIR/assign/monitors directory.

EXECUTION CONDITIONS

The on-line server must be started up.

DESCRIPTION OF THE CONFIGURATION

File: pacwst.ini

```
- FPARAM        : workstation configuration file
- PAC7CN        : conversation backup file
- TRACE         : trace activation option
                  (YES = activation, NO = no activation)
```

File: pacwstsock.ini

```
- FPARAM        : workstation configuration file
- PAC7CN        : conversation backup file
- TRACE         : trace activation option
                  (YES = activation, NO = no activation)
- PAC_HOST_SERVER_NAME   : name of the UNIX machine where
                           the on-line server is installed
- PAC_HOST_SERVER_SOCKET : number of the on-line server
                           communication socket
```

WORKSTATION CONFIGURATION

The workstation configuration is managed in the same way as for the PARM transaction.

See subchapter "START-UP OF THE PARM TRANSACTION".

2.5. PACLINK: STARTING A WINDOWS 'DUMB' WORKSTATION

START-UP OF A 'DUMB' WORKSTATION ON WINDOWS (PACLINK)

For MS-WINDOWS and WIN-OS/2 the user double clicks on the icon created during installation.

REMEMBER

1. The user must be connected to the local network, in order to access either the on-line server communication files, with communication via files, or the TCP/IP of the UNIX machine with communication via a socket.

2. The assignment of communication files is done in the GSPACXxx.PRM file, that of the UNIX machine and the associated communication socket in the GSTCPxx.PRM file.

   (xx represents the code specific to UNIX hardware).

   The user must update the corresponding file, in order to adapt the workstation configuration to that of the on-line server to which he/she wants to connect.

3. The communication files are created by the on-line server in the $PACDIR/commun directory. Their coding on the disk is made up of the on-line server name and the suffixes ".cm" and ".cd".

```
Example: J:\sertp1.cm
         J:\sertp1.cd

with:    J: = unit of common disk,
          sertp1 = on-line server name
```

   The communication socket associated to the on-line server is created when the on-line server is created on the UNIX machine.

NOTES ON THE USE OF WINDOWS 'DUMB' WORKSTATIONS

The user 'dumb' workstation on WINDOWS uses the GSPACLAN and PACLINK programs.

3. DATABASE MANAGEMENT UTILITIES

3.1. MLIB: DATABASE MANAGEMENT

3.1.1. MLIB: INTRODUCTION

MLIB: INTRODUCTION

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

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

. Create or delete libraries in an existing database.

EXECUTION CONDITIONS

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

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

ABENDS

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

3.1.2. MLIB: INPUT - PROCESSING - RESULTS

MLIB : INPUT-PROCESSING-RESULTS

USER INPUT

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

There are two types of specific user input:

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

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

The structure of the heading line is as follows:

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

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

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

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

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

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

UPDATE RULES

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

1. DELETION TRANSACTIONS:

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

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

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

2. CREATION TRANSACTIONS:

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

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

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

3. MODIFICATION TRANSACTIONS:

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

Structure loops are detected by the system.

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

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

PRINTED REPORTS

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

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

RESULTS

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

WARNING

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

This procedure increments the session number.

3.1.3. MLIB: DESCRIPTION OF STEPS

MLIB: DESCRIPTION OF STEPS

DATABASE VALIDATION: PTU100

This program is always executed.

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Printing-generation request file
  (in input-output if no simulation)
  PAC7AG
 -Error message file
  (in input-output)
  PAC7AE

.Input transaction file:
 -Update transactions
  PAC7MB

.Output files:
 -Sequential image of data
  PAC7RP
  (must have capacity for all data)
 -Sequential image of indexes
  PAC7NA
  (must have capacity for all indexes)
 -Sequential image of unsorted indexes
  PAC7NB
 -Temporary storage
  PAC7RQ
  (1 record)

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

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

```
.Return codes:
 - 0:  OK without simulation
 - 4:  OK with simulation
 - 8:  Error on the '*'-line (unauthorized user or
       error on input transactions)
 -12:  Error upon accessing database
```

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

SEQUENTIAL-IMAGE FORMATTING: PTU120

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

.Internal sort files
  Not assigned

.Permanent input files:
 -Data file
  (in input-output to update session number)
  PAC7AR

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

.Output file:
 -Sequential image of the database
  PAC7PC
 If Dispatch backup option:
 -Database sequential image 2
  PAC7PD

.Output reports:
 -None.
```

End of procedure without simulation

```
.Deletion of temporary files NA, NB, RP, and RQ
.Call of PCBAKUP file.
```

3.1.4. MLIB: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) MLIB BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        MLIB PROCEDURE"
echo "                        ==============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : DATABASE MANAGMENT
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# * .HEADING LINE (REQUIRED)
# *  COL  2   : 'G'
# *  COL  3   : 'I' TO INITIALIZE NEW DATABASE, OTHERWISE BLANK
# *  COL  4   : 'S' TO SIMULATE NETWORK UPDATE
# *           : ' ' FOR ACTUAL UPDATE
# *  DETAIL LINE (ONE FOR EACH LIBRARY MODIFICATION)
# *  COL  1   : TRANSACTION CODE (C, M OR D)
# *  COL  2   : '*'
# *  COL  3-5 : CODE OF THE LIBRARY TO BE CREATED, OR
# *           : CODE OF THE LIBRARY TO DELETE, OR
# *           : CODE OF THE LIBR. WHOSE UPPER LEVEL LIBRARY
# *           : IS TO BE MODIFIED
# *  COL  6-8 : UPPER LEVEL LIBRARY CODE
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AG.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBMLIB'
export PAC7MB
PAC7NA=$PACTMP'NA'
export PAC7NA
PAC7NB=$PACTMP'NB'
export PAC7NB
PAC7RP=$PACTMP'RP'
export PAC7RP
PAC7RQ=$PACTMP'RQ'
export PAC7RQ
PAC7EU=$PACTMP'MLIBEU.100'
export PAC7EU
PAC7EV=$PACTMP'MLIBEV.100'
export PAC7EV
PAC7DD=$PACTMP'MLIBDD.100'
export PAC7DD
echo "Execution : PTU100"
cobrun PTU100
```

```
    RETURN=$?
    case $RETURN in
    0)
     . $PACDIR/assign/$1/PAC7AR.ini
     . $PACDIR/assign/$1/PACSAVPC.ini
     PAC7PC=$PACSAVPCNEW
     export PAC7PC
     PAC7PD=$PACSAVPCINEW
     export PAC7PD
     PAC7AN=$PACTMP'NA'
     export PAC7AN
     PAC7NB=$PACTMP'NB'
     export PAC7NB
     PAC7PR=$PACTMP'RP'
     export PAC7PR
     PAC7PQ=$PACTMP'RQ'
     export PAC7PQ
     echo "Execution : PTU120"
     cobrun PTU120
     RETURN=$?
     case $RETURN in
     0)
      echo "End of procedure (without simulation)"
      echo ""
      echo "Call the file PCBACKUP.ini"
      sh $PACDIR/assign/$1/PCBACKUP.ini
      echo ""
      echo "Deletion of the temporary files"
      rm -f $PACTMP'NA'
      rm -f $PACTMP'NB'
      rm -f $PACTMP'RP'
      rm -f $PACTMP'RQ'
      ;;
     *)
      echo "Error in executing PTU120"
      ;;
     esac
     ;;
    12)
     echo "Error in executing PTU100"
     echo "Error 12 : error in database access"
     ;;
    8)
     echo "Error in executing PTU100"
     echo "Error 8 : error in transactions, or on * input line"
     ;;
    4)
     echo "End of procedure whith simulation"
     ;;
    *)
     echo "Error in executing PTU100"
     ;;
    esac
    if [ "$RETURN" != '0' -a "$RETURN" != '4' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

3.2. SAVE: DATABASE BACKUP

3.2.1. SAVE: INTRODUCTION

SAVE: INTRODUCTION

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

The back-up is performed on the following files:

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

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

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

EXECUTION CONDITIONS

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

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

ABENDS

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

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

ARCHIVAL AND BACKUP LINKING

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

. 0 : No error.

. 8 : Database not available

SIMPLIFIED BACKUP

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

3.2.2. SAVE: PROCESSING - RESULTS

SAVE: INPUT-RESULTS

PRINTED REPORT

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

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

- Two optional reports:

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

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

USER INPUT

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

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

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

The structure of the line is as follows:

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

OUTPUT

The output of the SAVE procedure is the following:

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

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

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

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

NOTES:

The SAVE procedure increments the current session number.

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

3.2.3. SAVE: DESCRIPTION OF STEPS

SAVE: DESCRIPTION OF STEPS

DATABASE CONSISTENCY CHECK: PTUBAS

```
.Permanent input files:
 -Data file
  PAC7AR
 -Error message file
  PAC7AE
 -Update serialization file
  PAC7LO

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

.Return code(s):
 -0:  OK.
 -4:  Database invalid, STOP triggered.
```

BACKUP OF THE DATABASE: PTU500

```
.Permanent input then input-output file:
 -Data file
  PAC7AR

.Permanent input files:
 -Error message file
  PAC7AE
 -Index File
  PAC7AN

.Input transaction file:
 -User transaction
  PAC7MB (MBSAVE file in INPUT directory)
```

```
.Output file:
 -Sequential image of the database
  PAC7PC (PC.NEW in the directory SAVE)
 If backup Dispatch option:
 -Sequential image 2 of the database
  PAC7PD (PCI.NEW in the directory SAVE of the database,
   created if OPTION: database backup on two files)

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

Return code(s):
- 8:   Database inconsistency or
       no batch-procedure authorization
```

Response to return code:

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

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

3.2.4. SAVE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SAVE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                        SAVE PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : DATABASE BACKUP
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# * ONE '*' LINE WITH USER CODE AND PASSWORD
# * .OPTIONAL REPORT INPUT
# * COL  2     : 'OR'
# * COL  8     : ' ' VA Pac STATISTICS PRINTING
# *            : 'N' NO PRINTING OF STATISTICS
# * COL  9     : ' ' VA Pac LIMITATIONS PRINTING
# *            : 'N' NO PRINTING OF VA Pac LIMITATIONS
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7DS=$PACTMP'SAVEDS.BAS'
export PAC7DS
. $PACDIR/assign/$1/SEMLOCK.ini
echo "Execution : PTUBAS"
rtscgi PTUBAS
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7AN.ini
 . $PACDIR/assign/$1/PAC7AR.ini
 PAC7MB=$PACINPUT'MBSAVE'
 export PAC7MB
 . $PACDIR/assign/$1/PACSAVPC.ini
 PAC7PC=$PACSAVPCNEW
 export PAC7PC
 PAC7PD=$PACSAVPCINEW
 export PAC7PD
 PAC7EU=$PACTMP'SAVEEU.500'
 export PAC7EU
 PAC7DS=$PACTMP'SAVEDS.500'
 export PAC7DS
 PAC7DD=$PACTMP'SAVEDD.500'
 export PAC7DD
 echo "Execution : PTU500"
 cobrun PTU500
 RETURN=$?
```

```
 case $RETURN in
 0)
  echo "End of procedure"
  echo ""
  echo "Call the file PCBACKUP.ini"
  sh $PACDIR/assign/$1/PCBACKUP.ini
  ;;
 8)
  echo "Error in executing PTU500"
  echo "Error 8 : Error on * input line"
  ;;
 *)
  echo "Error in executing PTU500"
  ;;
 esac
 ;;
4)
 echo "Error in executing PTUBAS"
 echo "Error 4 : database unavailable"
 ;;
*)
 echo "Error in executing PTUBAS"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

3.3. SASY: DATABASE SYSTEM BACKUP COMPLEMENT

3.3.1. SASY: INTRODUCTION

SASY : INTRODUCTION

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

The following files are to be backed up:

. Data file (AR),

. Index file (AN).

EXECUTION CONDITIONS

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

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

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

ABEND

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

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

USER INPUT

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

RESULT

This procedure increments the current session number.

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

3.3.2. SASY: DESCRIPTION OF STEPS

SASY: DESCRIPTION OF STEPS

DATABASE CONSISTENCY CHECK: PTUBAS

```
.Permanent input files:
 -Data file
  PAC7AR
 -Error message file
  PAC7AE
 -Update serialization file
  PAC7LO

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

.Return code(s):
 -0:  OK.
 -4:  Database invalid, STOP triggered.
```

SESSION NUMBER INCREMENTATION: PTU502

```
.Permanent input-output file:
 -Data file
  PAC7AR

.Permanent input file:
 -Error message file
  PAC7AE

.Output Report:
 -Review
  PAC7GZ
```

3.3.3. SASY: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SASY BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        SASY PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : DATABASE SYSTEM BACKUP
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7DS=$PACTMP'SASYDS.BAS'
export PAC7DS
. $PACDIR/assign/$1/SEMLOCK.ini
echo "Execution : PTUBAS"
rtscgi PTUBAS
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7AR.ini
 PAC7GZ=$PACTMP'SASYGZ.502'
 export PAC7GZ
 echo "Execution : PTU502"
 cobrun PTU502
 RETURN=$?
 case $RETURN in
 0)
  echo "End of procedure"
  ;;
 *)
  echo "Error in executing PTU502"
  ;;
 esac
 ;;
4)
 echo "Error in executing PTUBAS"
 echo "Database unavailable"
 ;;
*)
 echo "Error in executing PTUBAS"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

3.4. REST: DATABASE RESTORATION

3.4.1. REST: INTRODUCTION

REST: INTRODUCTION

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

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

EXECUTION CONDITIONS

The database must be closed to on-line processing.

Since this procedure re-creates the database, it is recommended to have previously readjusted the sizes of the different database files according to their estimated evolution. These modifications must be made in the System Parameter library.

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

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

ABNORMAL EXECUTION

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

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

3.4.2. REST: USER INPUT

REST : USER INPUT

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

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

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

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

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

(1): This date is used:

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

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

"SYSTEM EXPIRATION DATE".

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

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

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

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

NOTES:

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

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

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

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

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

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

SIMPLIFIED RESTORATION

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

OUTPUT REPORTS

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

GENERAL RESULTS

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

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

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

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

3.4.3. REST: DESCRIPTION OF STEPS

REST: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

.Input file:
CARTE

.Output file:
PAC7MB

.Permanent input file:
-Error message file
PAC7AE

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

.Return code(s):
-8: Unauthorized user

VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed only if the Journal file exists.

.Permanent input files:
-Error message file
PAC7AE
-Journal file
PAC7AJ

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

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

RESTORATION OF THE DATABASE: PTU400

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

.Permanent input files:
 -Error message file
  PAC7AE
 -Sequential image of the database
  PAC7PC
 If backup option Dispatch:
 -Sequential image of database #2
  PAC7PD (PCI in the directory SAVE)

.Permanent output files:
 -Data file
  PAC7AR
 -Index File
  PAC7AN
 -Journal file
  PAC7AJ

.Input transaction file:
 -User transactions
  PAC7MB

.Output file:
 -Working file (2 records)
  PAC7PS

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

DATABASE AVAILABILITY - TRANSACTION RETRIEVAL: PTU420

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

.Input-output file:
 -Data file
  PAC7AR

.Permanent input files:
 -Journal to apply
  PAC7JO (PJ in the directory SAVE)
 -Error message file
  PAC7AE

.Input work file:
 PAC7PS (PS in the temporary files directory)

.Output file:
 -Update transactions
  PAC7OJ

.Output report:
 -Retrieval report
  PAC7EU

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

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

DATABASE UPDATE: PACA15

.Permanent update files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Journal file
  PAC7AJ
 -Update serialization
  PAC7LO

.Permanent input files:
 -Error message file
  PAC7AE
 -DSMS file of VA Pac elements
  PAC7DC
  (DSM variant only)

.Input transaction file:
 -Update transactions
  PAC7MV (MV in the temporary files directory)

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

.Return code(s):
 - 0: OK without error
 - 2: Warning error
 - 4: Serious error

3.4.4. REST: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) REST BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        REST PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PAC7AJ.ini
echo "Directory of the AJ file          : `dirname $PAC7AJ.`"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : DATABASE RESTORATION
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# * '*' LINE WITH USER CODE AND PASSWORD
# * .RESTORATION PARAMETERS
# *  COL  2    : 'Y'
# *  COL  3-7  : NUMBER OF GAPS IN ABSOLUTE VALUE
# *  COL  8-9  : NUMBER OF GAPS IN PERCENTAGE (/ DATABASE)
# *  COL 10    : INITIAL LANGUAGE CODE (F=FRENCH, E=ENGLISH)
# *  COL 11    : '1' INHIBITION OF TRANSACTION LOG
# *  COL 12    : SYSTEM DATE FORMAT ('N' FOR  DD/MM/YY)
# *            :                    ('I' FOR  MM/DD/YY)
# *  COL 13-15 : 'REC' TO RECOVER ARCHIVED TRANSACTIONS
# *  COL 16-19 : 4 CHARACTERS APPEARING IN TOP RIGHT CORNER OF
# *            : VA Pac SCREENS
# *  COL 20-22 : MAXIMUM ACCESS NUMBER OF ON-LINE SEARCHES
# *            :  IN DATABASE (LISTS) (NNN) - (DEFAULT : 300)
# *  COL 23    : 'U' DEFAULT OPTION: IMPLICIT UPDATE
# *            : 'N' EXPLICIT UPDATE
# *  COL 35-46 : PFKEYS ASSIGNED FUNCTIONS
# *  COL 79    : 'D' SEQUENTIAL BACKUP OF THE DATABASE ON TWO
# *            : FILES
# * IF NO INPUT IS ENTERED, THE NUMBER OF EXISTING GAPS, AND
# * OTHER CHARACTERISTICS OF THE DATABASE, ARE UNCHANGED BY
# * THE PROCEDURE.
# *
# * IF THE DISK TRANSACTION JOURNAL FILE (AJ) IS NOT
# * REINITIALIZED, THE RESTORATION PROCEDURE IS NOT EXECUTED AND
# * IT IS NECESSARY TO EXECUTE THE ARCH PROCEDURE FIRST.
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
CARTE=$PACINPUT'MBREST'
export CARTE
PAC7MB=$PACTMP'MB'
export PAC7MB
PAC7DD=$PACTMP'RESTDD.004'
export PAC7DD
echo "Execution: PTU004"
```

```
    cobrun PTU004
    RETURN=$?
    case $RETURN in
    0)
     ;;
    8)
     echo "Error in executing PTU004"
     echo "Error 8: Unauthorized user"
     sh $PACDIR/batch/proc/ERRPAUSE.ini
     exit $RETURN
     ;;
    *)
     echo "Error in executing PTU004"
     echo "Error $RETURN"
     sh $PACDIR/batch/proc/ERRPAUSE.ini
     exit $RETURN
     ;;
    esac
    # ************************************************************
    . $PACDIR/assign/$1/PAC7AJ.ini
    if [ -r "$PAC7AJ" ]
    then
       . $PACDIR/assign/$1/PAC7AE.ini
       PAC7EU=$PACTMP'RESTEU.380'
       export PAC7EU
       echo "Execution : PTU380"
       cobrun PTU380
       RETURN=$?
       case $RETURN in
       0)
        ;;
       8)
        echo "Error in executing PTU380"
        echo "Error 8 : journal has not been archived"
        sh $PACDIR/batch/proc/ERRPAUSE.ini
        exit $RETURN
        ;;
       *)
        echo "Error in executing PTU380"
        sh $PACDIR/batch/proc/ERRPAUSE.ini
        exit $RETURN
        ;;
       esac
    fi
    . $PACDIR/assign/$1/PAC7AE.ini
    . $PACDIR/assign/$1/PAC7AJ.ini
    . $PACDIR/assign/$1/PAC7AN.ini
    . $PACDIR/assign/$1/PAC7AR.ini
    PAC7MB=$PACINPUT'MBREST'
    export PAC7MB
    . $PACDIR/assign/$1/PACSAVPC.ini
    PAC7PC=$PACSAVPC
    export PAC7PC
    PAC7PD=$PACSAVPCI
    export PAC7PD
    PAC7PS=$PACTMP'PS'
    export PAC7PS
    PAC7EU=$PACTMP'RESTEU.400'
    export PAC7EU
    PAC7DD=$PACTMP'RESTDD.400'
    export PAC7DD
    echo "Execution : PTU400"
    cobrun PTU400
    RETURN=$?
    case $RETURN in
    0)
     . $PACDIR/assign/$1/PAC7AE.ini
     . $PACDIR/assign/$1/PAC7AR.ini
     . $PACDIR/assign/$1/PACSAVPJ.ini
     PAC7JO=$PACSAVPJ
     export PAC7JO
     PAC7OJ=$PACTMP'OJ'
     export PAC7OJ
```

```
 PAC7PS=$PACTMP'PS'
 export PAC7PS
 PAC7EU=$PACTMP'RESTEU.420'
 export PAC7EU
 echo "Execution : PTU420"
 cobrun PTU420
 RETURN=$?
 case $RETURN in
 0)
  . $PACDIR/assign/$1/PAC7AE.ini
  . $PACDIR/assign/$1/PAC7AJ.ini
  . $PACDIR/assign/$1/PAC7AN.ini
  . $PACDIR/assign/$1/PAC7AR.ini
  . $PACDIR/assign/$1/PAC7DC.ini
  PAC7MV=$PACTMP'OJ'
  export PAC7MV
  PAC7IE=$PACTMP'RESTIE.A15'
  export PAC7IE
  PAC7IF=$PACTMP'RESTIF.A15'
  export PAC7IF
  . $PACDIR/assign/$1/SEMLOCK.ini
  echo "Execution : PACA15"
  rtscgi PACA15
  RETURN=$?
  case $RETURN in
  0)
   echo "End of procedure"
   echo ""
   echo "Deletion of the temporary files"
   rm -f $PACTMP'PS'
   rm -f $PACTMP'OJ'
   ;;
  2)
    echo "Error in executing PACA15"
    echo "Error 2 : At least one transaction with warning"
    sh $PACDIR/batch/proc/ERRPAUSE.ini
    exit $RETURN
   ;;
  4)
    echo "Error in executing PACA15"
    echo "Error 4 : At least one transaction is rejected"
    sh $PACDIR/batch/proc/ERRPAUSE.ini
    exit $RETURN
   ;;
  *)
   echo "Error in executing PACA15"
   ;;
  esac
  ;;
 4)
  echo "No transaction to be retrieved"
  echo "End of procedure"
  echo ""
  echo "Deletion of the temporary files"
  rm -f $PACTMP'PS'
  rm -f $PACTMP'OJ'
  ;;
 *)
  echo "Error in executing PTU420"
  ;;
 esac
 ;;
8)
 echo "Error in executing PTU400"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PTU400"
 ;;
esac
if [ "$RETURN" != '0' -a "$RETURN" != '4' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
```

```
      exit $RETURN
   else
      exit 0
   fi
```

3.5. RESY: DATABASE SYSTEM RESTORATION COMPLEMENT

3.5.1. RESY: INTRODUCTION

RESY: INTRODUCTION

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

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

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

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

EXECUTION CONDITIONS

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

On-line access must be closed.

ABEND

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

PRINTED RESULTS

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

NOTES:

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

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

3.5.2. RESY: USER INPUT - RESULTS

RESY : USER INPUT-RESULTS

USER INPUT

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

The input has the following structure:

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

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

(1): This date does the following:

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

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

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

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

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

The default option for the language code is French.

The number of gaps cannot be specified by this procedure.

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

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

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

3.5.3. RESY: DESCRIPTION OF STEPS

RESY: DESCRIPTION OF STEPS

VALIDATION OF JOURNAL CONTENTS: PTU380

This step is executed only if the Journal file exists.

.Permanent input files:
 -Error message file
  PAC7AE
 -Journal file
  PAC7AJ

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

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

DATABASE POSITIONING: PTU402

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

.Permanent output file:
 -Data file
  PAC7AR

.Permanent input file:
 -Error message file
  PAC7AE

.Input transaction file:
 -User transaction
  PAC7MB

. Output file:
 -Work file (2 recs.)
  PAC7PS

.Output report:
 -Restoration report
  PAC7GZ

DATABASE AVAILABILITY - TRANSACTION RETRIEVAL: PTU420

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

.Input-output file:
-Data file
PAC7AR

.Permanent input files:
-Journal to apply
PAC7JO (PJ in the directory SAVE)
-Error message file
PAC7AE

.Input work file:
PAC7PS (PS in the temporary files directory)

.Output file:
-Update transactions
PAC7OJ

.Output report:
-Retrieval report
PAC7EU

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

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

DATABASE UPDATE: PACA15

.Permanent update files:
-Data file
PAC7AR
-Index file
PAC7AN
-Journal file
PAC7AJ
-Update serialization
PAC7LO

.Permanent input files:
-Error message file
PAC7AE
-DSMS file of VA Pac elements
PAC7DC
(DSM variant only)

.Input transaction file:
-Update transactions
PAC7MV (MV in the temporary files directory)

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

.Return code(s):
- 0: OK without error
- 2: Warning error
- 4: Serious error

3.5.4. RESY: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) RESY BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        RESY PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PAC7AJ.ini
echo "Directory of the AJ file          : »dirname $PAC7AJ.»"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : DATABASE SYSTEM RESTORATION
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .RESTORATION PARAMETERS
# *  COL  2    : 'Y'
# *  COL 10    : INITIAL LANGUAGE CODE (F=FRENCH, E=ENGLISH)
# *  COL 11    : '1' INHIBITION OF TRANSACTION LOG
# *  COL 12    : SYSTEM DATE FORMAT ('N': DD/MM/YY)
# *            :                    ('I': MM/DD/YY)
# *  COL 13-15 : 'REC' FOR RECOVERY OF ARCHIVED TRANSACTIONS
# *  COL 16-19 : 4 CHARACTERS APPEARING AT TOP RIGHT OF VA Pac
# *            : SCREENS
# *  COL 20-22 : MAXIMUM ACCESS NUMBER OF ON-LINE SEARCHES IN
# *              DATABASE (LISTS) (NNN) - (DEFAULT VALUE: 300)
# *  COL 23    : 'U' (DEFAULT OPTION): IMPLICIT UPDATE
# *            : 'N' EXPLICIT UPDATE
# *  COL 35-46 : PFKEYS ASSIGNED FUNCTIONS
# *  COL 79    : 'D' SEQUENTIAL BACKUP OF THE DATABASE ON TWO
# *            : FILES
# * IF THE DISK TRANSACTION JOURNAL FILE (AJ) IS NOT REINITIA-
# * LIZED, THE RESTORATION PROCEDURE IS NOT EXECUTED. THE
# * ARCH PROCEDURE MUST THEN BE EXECUTED.
# ************************************************************
. $PACDIR/assign/$1/PAC7AJ.ini
if [ -r "$PAC7AJ" ]
then
   . $PACDIR/assign/$1/PAC7AE.ini
   PAC7EU=$PACTMP'RESYEU.380'
   export PAC7EU
   echo "Execution : PTU380"
   cobrun PTU380
   RETURN=$?
   case $RETURN in
   0)
    ;;
   8)
    echo "Error in executing PTU380"
    echo "Error 8 : journal has not been archived"
    sh $PACDIR/batch/proc/ERRPAUSE.ini
```

```
        exit $RETURN
        ;;
       *)
        echo "Error in executing PTU380"
        sh $PACDIR/batch/proc/ERRPAUSE.ini
        exit $RETURN
        ;;
       esac
    fi
    . $PACDIR/assign/$1/PAC7AE.ini
    . $PACDIR/assign/$1/PAC7AR.ini
    PAC7MB=$PACINPUT'MBRESY'
    export PAC7MB
    PAC7PS=$PACTMP'PS'
    export PAC7PS
    PAC7GZ=$PACTMP'RESYGZ.402'
    export PAC7GZ
    echo "Execution : PTU402"
    cobrun PTU402
    RETURN=$?
    case $RETURN in
    0)
     . $PACDIR/assign/$1/PAC7AE.ini
     . $PACDIR/assign/$1/PAC7AR.ini
     . $PACDIR/assign/$1/PACSAVPJ.ini
     PAC7JO=$PACSAVPJ
     export PAC7JO
     PAC7OJ=$PACTMP'OJ'
     export PAC7OJ
     PAC7PS=$PACTMP'PS'
     export PAC7PS
     PAC7EU=$PACTMP'RESYEU.420'
     export PAC7EU
     echo "Execution : PTU420"
     cobrun PTU420
     RETURN=$?
     case $RETURN in
     0)
      . $PACDIR/assign/$1/PAC7AE.ini
      . $PACDIR/assign/$1/PAC7AJ.ini
      . $PACDIR/assign/$1/PAC7AN.ini
      . $PACDIR/assign/$1/PAC7AR.ini
      . $PACDIR/assign/$1/PAC7DC.ini
      PAC7MV=$PACTMP'OJ'
      export PAC7MV
      PAC7IE=$PACTMP'RESYIE.A15'
      export PAC7IE
      PAC7IF=$PACTMP'RESYIF.A15'
      export PAC7IF
      . $PACDIR/assign/$1/SEMLOCK.ini
      echo "Execution : PACA15"
      rtscgi PACA15
      RETURN=$?
      case $RETURN in
      0)
       echo "End of procedure"
       echo ""
       echo "Deletion of the temporary files"
       rm -f $PACTMP'PS'
       rm -f $PACTMP'OJ'
       ;;
      2)
        echo "Error in executing PACA15"
        echo "Error 2 : At least one transaction with warning"
        sh $PACDIR/batch/proc/ERRPAUSE.ini
        exit $RETURN
       ;;
      4)
        echo "Error in executing PACA15"
        echo "Error 4 : At least one transaction is rejected"
        sh $PACDIR/batch/proc/ERRPAUSE.ini
        exit $RETURN
       ;;
```

```
      *)
       echo "Error in executing PACA15"
       ;;
      esac
      ;;
     4)
      echo "No transaction to be retrieved"
      echo 'End of procedure'
      echo ""
      echo "Deletion of the temporary files"
      rm -f $PACTMP'PS'
      rm -f $PACTMP'OJ'
      ;;
     *)
      echo "Error in executing PTU420"
      ;;
     esac
     ;;
    *)
     echo "Error in executing PTU402"
    ;;
    esac
    if [ "$RETURN" != '0' -a "$RETURN" != '4' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
       exit $RETURN
    else
       exit 0
    fi
```

3.6. ARCH: JOURNAL ARCHIVAL

3.6.1. ARCH: INTRODUCTION

ARCH: INTRODUCTION

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

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

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

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

EXECUTION CONDITION

On-line access must be closed down.

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

ABENDS

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

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

3.6.2. ARCH: INPUT - RECOMMENDATIONS - RESULTS

ARCH: USER INPUT

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

This procedure includes specific optional input for:

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

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

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

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

The structure of this input is as follows:

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

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

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

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

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

In case of an error on one of the options, an error message is printed and the archive is generated using the default options.

RECOMMENDATIONS

If there is no user input, this procedure can only be executed if the Database is in a consistent state, and if the archived transaction file is correctly formatted.

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

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

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

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

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

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

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

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

PRINTED OUTPUT

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

RESULTS

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

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

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

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

3.6.3. ARCH: DESCRIPTION OF STEPS

ARCH: DESCRIPTION OF STEPS

PARTICULAR CASE OF THE FIRST DATABASE ARCHIVING

In order for the first database archival to run correctly, the PJ file, containing archived transactions used as input of the procedure, is created as an empty file, and stored in the \SAVE directory during installation.

ARCHIVED-TRANSACTION PURGE

When a purge of archives is requested in the transactions files, two situations are possible:

1. The user does not wish to keep the purged archives in the PJ file: the file with PAC7PQ as internal name must be assigned to 'NUL'.

   This is done as a default in the procedure command file.

   ```
   PAC7PQ=/dev/null
   ```

2. The user wishes to keep purged archives in the PJ file: the file with PAC7PQ as internal name must be assigned, and it must correspond to a disk file.

   In this case, modify the procedure command file according to the following example:

   ```
   SET PAC7PQ=%6:%1 SAVE %2 PQ

   PAC7PQ=$PACDIR/save/$1/PQ
   ```

In this case, the parameter %8 will no longer be used in the procedure.

ARCHIVAL OF JOURNAL FILE: PTU300

This step:

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

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

. Updates the file of archived transactions.

```
.Permanent input files:
 -Error message file
  PAC7AE
 -Previously archived transactions
  PAC7JP
 -Journal file to reinitialize
  PAC7AJ

.Input work file:
 -User transaction
  PAC7MB

.Permanent input-Output file:
 -Data file
  PAC7AR

.Output files:
 -Archived update transactions
  PAC7PJ
 -Deactivated transactions
  PAC7PQ (Assigned to save deactivaed transactions)
  The DSN must be entered in order to keep these deactivated
  transactions.

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

.Return codes:
 . 0:  No error detected on the files,
 . 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 (MBARCH).
 .12:  Input-output error on a file.
```

RE-INITIALIZATION OF THE JOURNAL FILE: PTU320

This step executes the following:

- Creates the first record in the Journal file,

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

.Input work file:
-User transaction
PAC7MB (MBARCH file in INPUT directory)

.Permanent input/output file:
-Data file
PAC7AR

.Permanent input file:
-Error message file
PAC7AE

.Output file:
-Journal file to re-initialize
PAC7AJ

.Output report:
-Review of reinitialization
PAC7EU

.Return codes:
0: No error detected,
8: The database is not available.

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

3.6.4. ARCH: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) ARCH BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        ARCH PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PAC7AJ.ini
echo "Directory of the AJ file          : »dirname $PAC7AJ.»"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TRANSACTIONS ARCHIVING
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# * '*' LINE WITH USER CODE AND PASSWORD
# * .ARCHIVED TRANSACTIONS DEACTIVATION COMMAND
# *  COL  2    : 'S'
# *  COL  3-6  : SESSION NUMBER
# *  COL  7-14 : DATE (CCYYMMDD)
# *  COL 15    : ' ' ARCHIVED TRANSACTION FILE AVAILABLE
# *            : 'I' ABSENCE OF ARCHIVED TRANSACTION FILE
# *  COL 16    : ' ' DATA FILE AVAILABLE
# *            : 'D' ABSENCE OF DATA FILE
# *  COL 17    : ' ' ARCHIVING AND RE-INITIALIZATION
# *            : 'J' RE-INITIALIZATION WITHOUT ARCHIVING
# *
# * IF NO COMMAND IS ENTERED OR IF THERE IS AN INVALID PARAME-
# * TER, DEACTIVATION DOES NOT HAPPEN, BUT ARCHIVING AND RE-
# * INITIALIZATION WILL TAKE PLACE NORMALLY.
# *
# * TRANSACTIONS WHOSE SESSION (DATE) IS PRIOR OR EQUAL TO THE
# * SESSION (DATE) ENTERED WILL NOT BE ARCHIVED. THEY WILL BE
# * RETREIVED IN THE DEACTIVATED TRANSACTION FILE INSTEAD.
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AJ.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBARCH'
export PAC7MB
. $PACDIR/assign/$1/PACSAVPJ.ini
PAC7JP=$PACSAVPJ
export PAC7JP
PAC7PJ=$PACSAVPJNEW
export PAC7PJ
. $PACDIR/assign/$1/PACSAVPQ.ini
PAC7PQ=$PACSAVPQ
export PAC7PQ
PAC7EU=$PACTMP'ARCHEU.300'
export PAC7EU
```

```
    PAC7DD=$PACTMP'ARCHDD.300'
    export PAC7DD
    echo "Execution : PTU300"
    cobrun PTU300
    RETURN=$?
    case $RETURN in
    0)
     . $PACDIR/assign/$1/PAC7AE.ini
     . $PACDIR/assign/$1/PAC7AJ.ini
     . $PACDIR/assign/$1/PAC7AR.ini
     PAC7MB=$PACINPUT'MBARCH'
     export PAC7MB
     PAC7EU=$PACTMP'ARCHEU.320'
     export PAC7EU
     echo "Execution : PTU320"
     cobrun PTU320
     RETURN=$?
     case $RETURN in
     0)
      echo "End of procedure"
      echo ""
      echo "Call the file PJBACKUP.ini"
      sh $PACDIR/assign/$1/PJBACKUP.ini
      ;;
     8)
      echo "Error in executing PTU320"
      echo "Error 8 : database unavailable"
      ;;
     *)
      echo "Error in executing PTU320"
      ;;
     esac
     ;;
    12)
     echo "Error in executing PTU300"
     echo "Error 12 : input-output on a file"
     ;;
    8)
     echo "Error in executing PTU300"
     echo "Error 8 : Error on * input line"
     echo "       or database unavailable"
     ;;
    *)
     echo "Error in executing PTU300"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

3.7. REOR: DATABASE REORGANIZATION

3.7.1. REOR: INTRODUCTION

REOR: INTRODUCTION

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

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

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

The REOR procedure may be used in two cases:

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

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

- Obsolete libraries and/or sessions;

- Entities not used in the database;

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

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

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

. Deletion of unused Production sessions (PEI Function)

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

EXECUTION CONDITION

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

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

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

ABENDS

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

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

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

3.7.2. REOR: INPUT - RECOMMENDATIONS

REOR: USER INPUT

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

Specific user input for the procedure (optional), specifying

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

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

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

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

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

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

Maximum number of occurrences by type: 2,500.

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

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

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

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

ESTIMATING FILE SIZE

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

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

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

PRINTED OUTPUT

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

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

RESULTS

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

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

IMPORTANT RECOMMENDATIONS

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

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

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

3.7.3. REOR: DESCRIPTION OF STEPS

REOR: DESCRIPTION OF STEPS

DISK SPACE REQUIRED BY SORT PROGRAMS

The REOR procedure includes two SORT programs:

. PTU205 sorts data, i.e., the PR temporary file created by the PTU200 program,

. PTU225 sorts indexes, i.e., the AN temporary file created by the PTU220 program.

Each SORT programs requires disk space equivalent to twice the size of the file to be sorted. This space must be allocated on the disk from which the procedure is run (default: "release"\BATCH\PROC). You can modify this allocation with the command: SET TMP=... files.

VALIDATION OF USER INPUT: PTU2CL

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

```
.Permanent input files:
 -Error message file
  PAC7AE

.Input work file:
  PAC7MB

.Output file:
 -Formatted records
  PAC7BM

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

.Return code(s):
  0:  OK
  4:  Error on user input
  8:  Unauthorized user.
```

RETRIEVAL OF DATA: PTU200

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

```
.Permanent input files:
```

-Error message file
PAC7AE
-Sequential image of the database
PAC7PC

.Output file:
-Formatted records
PAC7PR (PR in the temporary files directory)

.Output reports:
-Retrieval statistics
PAC7EE

DATA SORT: PTU205

.Input file:
-Formatted records: PAC7PR
(PR file in the temp. dir.)

.Output file:
-Sorted records: PAC7RP
(RP file in the temp. dir.)

STEP END: deletion of PR file.

EXTRACTION FOR PURGE OF ENTITIES: PTU208

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

.Internal sort files:
Not assigned

.Input work file:
-User transactions
PAC7MB (MBREOR file in INPUT directory)

.Permanent input file:
-Error messages
PAC7AE

.Output file:
-Entity records to purge
PAC7PU (PU in the temporary files directory)

.Output report:
-Entity-purge transactions
PAC7EE

PURGE: PTU210

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

.Internal sort
Not assigned

.Input work files:
-Sorted records
PAC7PR (RP in the temporary files directory)
-Entity records to be purged
PAC7PU (PU in the temporary files directory)
-User transactions
PAC7MB (MBREOR file in INPUT directory)

.Permanent input file:
-Error message file
PAC7AE

.Output work files:
-Purged records
PAC7QS (QS in the temporary files directory)

```
 -Macro-Structure call lines
  PAC7UM (UM in the temporary files directory)

.Output reports:
 -Library and session purge report
  PAC7EE
 -Entity-purge report
  PAC7EK
 -Technical report
  PAC7EB

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

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

INDEX RECONSTRUCTION: PTU220

This step executes two types of procedures:

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

```
.Input work files:
 -Purged data
  PAC7UR
 -Macro-Structure call lines
  PAC7UM  (UM in the temporary-file directory)

.Permanent input file:
 -Error message file
  PAC7AE

.Output files:
 -Data from frozen sessions
  PAC7PA (PA in the temporary-file directory, size NH)
 -Data from the current session
  PAC7PB (PB in the temporary-file directory, size NC)
 -First data record
  PAC7PC (PCTEMP in the temporary-file directory)
 -Temporary index file
  PAC7AN (AN in the temporary-file directory, size NI)

.Work file (output, then input)
 -Macro-Structure call lines
  PAC7MR (MR in the temporary files directory)

.Output report:
 -Index-building report
  PAC7EE
```

SORT ON INDEXES: PTU225

```
.Input file:
 -Temporary index:  PAC7AN
  (AN file in the temp. dir.)

.Output file:
 -Sorted index:  PAC7NA
  (NA file in the temp. dir.)

STEP END: Deletion of AN file
```

MERGE: PTU240

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

.Permanent input file:
 -Error message file
  PAC7AE

.Input work files:
 -User transactions
  PAC7MB
 -Data from the frozen session
  PAC7PA (PA in the temporary-file directory)
 -Data from the current session
  PAC7PB (PB in the temporary-file directory)
 -First data record
  PAC7PC (PCTEMP in the temporary-file directory)
 -Sorted index file
  PAC7AN (NA in the temporary-file directory)

.Permanent output file:
 -Sequential image of the database
  PAC7CP (PC.NEW in directory SAVE)
 If Dispatch option of backup:
 -Sequential image of the database #2
  PAC7PD (PCI.NEW in directory SAVE)

.Output report:
 -Logical database building
  PAC7IE

3.7.4. REOR: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) REOR BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                        REOR PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : DATABASE REORGANIZATION
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# * .LIBRARY PURGE
# *  COL  2    : 'B' LIBRARY PURGE
# *  COL  3    : CODE OF LIBRARY TO BE PURGED (x23)
# *            : UP TO 23 CODES PER LINE
# * .SESSION PURGE
# *  COL  2    : 'V' FROZEN SESSION PURGE OR
# *            : 'S' SAVE FROZEN SESSIONS
# *  COL  3    : SESSION NUMBER (x17)
# *            : UP TO 17 SESSION NUMBERS PER LINE
# * .ENTITY PURGE
# *  COL  2    : 'E' ENTITY PHYSICAL PURGE
# *              (TRANSACTIONS PROVIDED BY EXPU)
# *  COL  3-5  : ENTITY TYPE
# *  COL  6-11 : CODE OF ENTITY TO BE PURGED
# *  COL 12-14 : LIBRARY CODE
# *            : TYPE/ENTITY/LIBRARY: UP TO 5 GROUPS PER LINE
# ************************************************************
# INPUT CONTROL
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBREOR'
export PAC7MB
PAC7BM=$PACTMP'MB'
export PAC7BM
PAC7EE=$PACTMP'REOREE.2CL'
export PAC7EE
PAC7DD=$PACTMP'REORDD.2CL'
export PAC7DD
echo "Execution : PTU2CL"
cobrun PTU2CL
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTU2CL"
 echo "Error $RETURN"
```

```
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
# * REPLACING LOW-VALUES BY SPACE
. $PACDIR/assign/$1/PACSAVPC.ini
PAC7MC=$PACSAVPC
export PAC7MC
PAC7PC=$PACTMP'PC'
export PAC7PC
echo "Execution : PTULVB"
echo data ¦ cobrun PTULVB
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTULVB"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
# * REORGANIZATION
. $PACDIR/assign/$1/PAC7AE.ini
PAC7PC=$PACTMP'PC'
export PAC7PC
PAC7PR=$PACTMP'PR'
export PAC7PR
PAC7EE=$PACTMP'REOREE.200'
export PAC7EE
PAC7DD=$PACTMP'REORDD.200'
export PAC7DD
echo "Execution : PTU200"
cobrun PTU200
RETURN=$?
case $RETURN in
0)
 rm -f $PACTMP'PC'
 PAC7PR=$PACTMP'PR'
 export PAC7PR
 PAC7RP=$PACTMP'RP'
 export PAC7RP
 case $PACSORT in
  unix ¦ UNIX) # unix sort
         echo "Unix sort of data file"
         cgiasc2ebc $PAC7PR $TMPDIR/PR.ebc
         sort -y +0.0 -0.21 +0.39 -0.43 +0.27 -0.31 +0.31 \
         -0.32 -r +0.45 -0.49 +0.21 -0.22 -r -T $TMPDIR \
         -o $TMPDIR/RP.ebc $TMPDIR/PR.ebc
         RETURN=$?
         rm -f $TMPDIR/PR.ebc
         cgiebc2asc $TMPDIR/RP.ebc $PAC7RP
         rm -f $TMPDIR/RP.ebc
         ;;
  *)           # cobol sort
         echo "Execution : PTU205"
         cobrun PTU205
         RETURN=$?
         ;;
 esac
 # File size control
 if [ "`cgisize $PAC7PR`" != "`cgisize $PAC7RP`" ]
 then
     RETURN=20
 fi
 case $RETURN in
 0)
  echo "Deletion of the temporary file :"
  echo $PACTMP'PR'
  rm -f $PACTMP'PR'
  . $PACDIR/assign/$1/PAC7AE.ini
```

```
PAC7MB=$PACTMP'MB'
export PAC7MB
PAC7PU=$PACTMP'PU'
export PAC7PU
PAC7EE=$PACTMP'REOREE.208'
export PAC7EE
echo "Execution : PTU208"
cobrun PTU208
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 PAC7MB=$PACTMP'MB'
 export PAC7MB
 PAC7PR=$PACTMP'RP'
 export PAC7PR
 PAC7PU=$PACTMP'PU'
 export PAC7PU
 PAC7UM=$PACTMP'UM'
 export PAC7UM
 PAC7QS=$PACTMP'QS'
 export PAC7QS
 PAC7EB=$PACTMP'REOREB.210'
 export PAC7EB
 PAC7EE=$PACTMP'REOREE.210'
 export PAC7EE
 PAC7EK=$PACTMP'REOREK.210'
 export PAC7EK
 echo "Execution : PTU210"
 cobrun PTU210
 RETURN=$?
 case $RETURN in
 0)
  echo "Deletion of the temporary files :"
  echo $PACTMP'RP'
  echo $PACTMP'PU'
  rm -f $PACTMP'RP'
  rm -f $PACTMP'PU'
  . $PACDIR/assign/$1/PAC7AE.ini
  PAC7MR=$PACTMP'MR'
  export PAC7MR
  PAC7UM=$PACTMP'UM'
  export PAC7UM
  PAC7UR=$PACTMP'QS'
  export PAC7UR
  PAC7AN=$PACTMP'AN'
  export PAC7AN
  PAC7PA=$PACTMP'PA'
  export PAC7PA
  PAC7PB=$PACTMP'PB'
  export PAC7PB
  PAC7PC=$PACTMP'PCTEMP'
  export PAC7PC
  PAC7EE=$PACTMP'REOREE.220'
  export PAC7EE
  echo "Execution : PTU220"
  cobrun PTU220
  RETURN=$?
  case $RETURN in
  0)
   echo "Deletion of the temporary files :"
   echo $PACTMP'MR'
   echo $PACTMP'UM'
   echo $PACTMP'QS'
   rm -f $PACTMP'MR'
   rm -f $PACTMP'UM'
   rm -f $PACTMP'QS'
   PAC7AN=$PACTMP'AN'
   export PAC7AN
   PAC7NA=$PACTMP'NA'
   export PAC7NA
   case $PACSORT in
    unix ¦ UNIX) # unix sort
```

```
               echo "Unix sort of index file"
               cgiasc2ebc $PAC7AN $TMPDIR/AN.ebc
               sort -y -T $TMPDIR -o $TMPDIR/NA.ebc $TMPDIR/AN.ebc
               RETURN=$?
               rm -f $TMPDIR/AN.ebc
               cgiebc2asc $TMPDIR/NA.ebc $PAC7NA
               rm -f $TMPDIR/NA.ebc
               ;;
        *)           # cobol sort
               echo "Execution : PTU225"
               cobrun PTU225
               RETURN=$?
               ;;
       esac
       # File size control
       if [ "`cgisize $PAC7AN`" != "`cgisize $PAC7NA`" ]
       then
           RETURN=20
       fi
       case $RETURN in
       0)
        echo "Deletion of the temporary file :"
        echo $PACTMP'AN'
        rm -f $PACTMP'AN'
        . $PACDIR/assign/$1/PAC7AE.ini
        . $PACDIR/assign/$1/PACSAVPC.ini
        PAC7CP=$PACSAVPCNEW
        export PAC7CP
        PAC7PD=$PACSAVPCINEW
        export PAC7PD
        PAC7AN=$PACTMP'NA'
        export PAC7AN
        PAC7PA=$PACTMP'PA'
        export PAC7PA
        PAC7PB=$PACTMP'PB'
        export PAC7PB
        PAC7PC=$PACTMP'PCTEMP'
        export PAC7PC
        PAC7IE=$PACTMP'REORIE.240'
        export PAC7IE
        PAC7MB=$PACTMP'MB'
        export PAC7MB
        echo "Execution : PTU240"
        cobrun PTU240
        RETURN=$?
        case $RETURN in
        0)
         echo "Deletion of the temporary files :"
         echo $PACTMP'NA'
         echo $PACTMP'PA'
         echo $PACTMP'PB'
         echo $PACTMP'PCTEMP'
         rm -f $PACTMP'NA'
         rm -f $PACTMP'PA'
         rm -f $PACTMP'PB'
         rm -f $PACTMP'PCTEMP'
         echo "End of procedure"
         echo ""
         echo "Call the file PCBACKUP.ini"
         sh $PACDIR/assign/$1/PCBACKUP.ini
         ;;
        *)
         echo "Error in executing PTU240"
         ;;
        esac
        ;;
       20)
        echo "Sort error"
        ;;
       *)
        echo "Error in executing PTU225"
        ;;
       esac
```

```
        ;;
       *)
        echo "Error in executing PTU220"
        ;;
       esac
       ;;
      8)
       echo "Error in executing PTU210"
       echo "Error 8 : capacity exceeded"
       ;;
      4)
       echo "Error in executing PTU210"
       echo "Error 4 : Error in input transactions"
       ;;
      *)
       echo "Error in executing PTU210"
       ;;
      esac
      ;;
     4)
      echo "Error in executing PTU208"
      echo "Error 4 : Error in input transactions"
      ;;
     *)
      echo "Error in executing PTU208"
      ;;
     esac
     ;;
    20)
     echo "Sort error"
     ;;
    *)
     echo "Error in executing PTU205"
     ;;
    esac
    ;;
   8)
    echo "Error in executing PTU200"
    echo "Error 8 : Error on * input line"
   ;;
   *)
    echo "Error in executing PTU200"
    ;;
   esac
   if [ "$RETURN" != '0' ]
   then
      sh $PACDIR/batch/proc/ERRPAUSE.ini
   fi
   exit $RETURN
```

3.8. SVAG: GENERATION-PRINT REQUEST BACKUP

3.8.1. SVAG: INTRODUCTION

SVAG: INTRODUCTION

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

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

EXECUTION CONDITION

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

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

ABEND

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

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

USER INPUT

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

3.8.2. SVAG: DESCRIPTION OF STEPS

SVAG: DESCRIPTION OF STEPS

BACKUP OF GENERATION-PRINTING REQUESTS: PTU550

.Input files:
 -Requests
  PAC7AG
 -Error messages
  PAC7AE
 -User input
  PAC7MB (MBSVAG file in INPUT directory)

.Output file:
 -Sequential image of requests
  PAC7PG (PG.NEW in SAVE directory)

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

.Return code:
  8: unauthorized user

3.8.3. SVAG: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SVAG BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        SVAG PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : AG FILE BACKUP
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *        '*' LINE WITH USER CODE AND PASSWORD
# ************************************************************
. $PACDIR/assign/$1/PAC7AG.ini
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBSVAG'
export PAC7MB
. $PACDIR/assign/$1/PACSAVPG.ini
PAC7PG=$PACSAVPGNEW
export PAC7PG
PAC7EE=$PACTMP'SVAGEE.550'
export PAC7EE
PAC7DD=$PACTMP'SVAGDD.550'
export PAC7DD
echo "Execution : PTU550"
cobrun PTU550
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 echo ""
 echo "Call the file PGBACKUP.ini"
 sh $PACDIR/assign/$1/PGBACKUP.ini
 ;;
8)
 echo "Error in executing PTU550"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PTU550"
;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

3.9. REAG: GENERATION-PRINT REQUEST RESTORATION

3.9.1. REAG: INTRODUCTION

REAG: INTRODUCTION

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

EXECUTION CONDITION

On-line access must be closed.

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

3.9.2. REAG: USER INPUT

REAG: USER INPUT

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

The procedure requires the following specific input (optional):

One line to specify the request:

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

One line per purge (in case of reorganization):

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

Maximum number of sessions.....: 500

Maximum number of libraries.....: 100

Maximum number of users.....: 100

Default option: restoration.

3.9.3. REAG: DESCRIPTION OF STEPS

REAG: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

.Input file:
CARTE

.Output file:
PAC7MB

.Permanent input file:
-Error message file
PAC7AE

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

.Return code(s):
-8: Unauthorized user

INITIALIZATION-REORGANIZATION OF REQUEST FILE (AG): PTU560

.Permanent input files:
-Sequential image of requests
PAC7PG
-Error message file
PAC7AE

.Permanent output file:
-Request file
PAC7AG

.Input transaction file:
-User transactions
PAC7MB

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

3.9.4. REAG: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) REAG BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        REAG PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : INITIALIZATION-RESTORATION OF AG FILE
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# * .RESTORATION OR INITIALIZATION REQUEST (REST. IF NO REQUEST)
# *  COL  2-3  : 'AG'
# *  COL  4    : ' ' RESTORATION
# *            : 'I' INITIALIZATION
# * .PURGE REQUEST (N OPTIONAL LINES)
# *  COL  2-6  : 'ABXXX' PURGE LIBRARY COMMANDS
# *  COL  2-7  : 'ASXXXX' PURGE SESSION COMMANDS
# *  COL  2-11 : 'AUXXXXXXXX' PURGE USER COMMANDS
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
CARTE=$PACINPUT'MBREAG'
export CARTE
PAC7MB=$PACTMP'MB'
export PAC7MB
PAC7DD=$PACTMP'REAGDD.004'
export PAC7DD
echo "Execution: PTU004"
cobrun PTU004
RETURN=$?
case $RETURN in
0)
 ;;
8)
 echo "Error in executing PTU004"
 echo "Error 8: Unauthorized user"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PTU004"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
```

```
. $PACDIR/assign/$1/PAC7AG.ini
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBREAG'
export PAC7MB
. $PACDIR/assign/$1/PACSAVPG.ini
PAC7PG=$PACSAVPG
export PAC7PG
PAC7EE=$PACTMP'REAGEE.560'
export PAC7EE
PAC7EK=$PACTMP'REAGEK.560'
export PAC7EK
PAC7DD=$PACTMP'REAGDD.560'
export PAC7DD
echo "Execution : PTU560"
cobrun PTU560
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
8)
 echo "Error in executing PTU560"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PTU560"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

3.10. PARM: UPDATE OF USER PARAMETERS

3.10.1. PARM: INTRODUCTION

PARM : INTRODUCTION

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

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

These user parameters may be updated in the following ways:

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

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

NOTES:

Some user parameters must be accessible on-line:

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

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

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

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

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

EXECUTION CONDITION

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

ABENDS

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

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

3.10.2. PARM: INPUT - RECOMMENDATIONS

PARM: USER INPUT

One line "*" (required):

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

There are two types of user input control lines:

1. FILE MANAGEMENT REQUESTS:

   Backup-reloading or restoration-reloading.

2. USER PARAMETER UPDATES:

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

1. FILE MANAGEMENT REQUESTS

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

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

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

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

IMPORTANT RECOMMENDATION

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

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

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

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

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

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

2. USER PARAMETERS

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

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

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

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

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

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

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

2.4 Correspondence table for special characters of keywords

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

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

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

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

2.6 Definition of methodology choices for the WorkStation

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

3.10.3. PARM: USER-CODE DEFINITION

DEFINITION OF USER CODES

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

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

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

1. GLOBAL AUTHORIZATION LEVEL

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

2. AUTHORIZATION LEVEL ASSOCIATED TO A VA PAC DATABASE

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

2. AUTHORIZATION LEVEL ASSOCIATED TO A DATABASE LIBRARY

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

LIBRARY ACCESS AUTHORIZATIONS

The authorization levels are:

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

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

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

NOTES:

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

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

Example:

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

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

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

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

Update of a library-authorization level

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

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

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

NOTES

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

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

USER-PARAMETER MANAGEMENT ACCESS AUTHORIZATION

The authorization levels are:

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

(See the explanation below.)

BATCH PROCEDURE ACCESS AUTHORIZATION (option)

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

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

PEI FUNCTION ACCESS AUTHORIZATION

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

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

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

3.10.4. PARM: USER-CODE GLOBAL AUTHORIZATIONS

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

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

3.10.5. PARM: USER-CODE SPECIFIC AUTHORIZATIONS

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

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

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

3.10.6. PARM: TEXT TYPES

PARM: TEXT TYPES

UPDATING TEXT TYPES

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

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

Updating includes creation, modification or deletion in the file.

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

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

3.10.7. PARM: MODIFICATION OF STANDARD ERROR MESSAGES

MODIFICATIONS OF STANDARD ERROR MESSAGES

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

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

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

NOTES

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

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

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

3.10.8. PARM: GENERATED-STREAM CONTROL CARDS

PARM: GENERATED-STREAM CONTROL CARDS

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

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

These job control commands have a two-fold purpose:

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

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

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

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

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

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

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

CALL OF CONTROL CARDS

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

The user must do the following:

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

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

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

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

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

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

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

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

PARAMETERIZATION OF CONTROL CARDS

Job control cards are parameterized according to the following principles:

A control card consists of three types of information:

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

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

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

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

It is specified in the line's last character.

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

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

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

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

EXAMPLE

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

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

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

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

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

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

.'P'= Generated program code.

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

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

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

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

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

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

3.10.9. PARM: DESCRIPTION OF STEPS

PARM: DESCRIPTION OF STEPS

IMPORTANT

The standard installation creates the AE and AP files in the BASES directory ('release'\BASES for VisualAge Pacbase OS2 or WINDOWS NT, and $SPACDIR/bases for VisualAge Pacbase UNIX), and the AEO and PE sequential backups in the SAVE directory ("release"\SAVE for VA Pac OS/2 or WINDOWS NT and $PACDIR/save for VA Pac UNIX) because these files can manage several VisualAge Pacbase databases.

The MBPARM transaction file is located in the 'release'\INPUT\'db_name' directory ('release'\INPUT\'db_name' for VA Pac OS/2 or WINDOWS NT, and $PACDIR/input/'db_name' for VA Pac UNIX).

In the case of a multi-database installation, the database administrator should therefore manage only one MBPARM file and run the PARM procedure always on the same database.

The PEBACKUP file, run at the end of the procedure when there is no error, manages the rotation of PE backups in the 'release'\SAVE directory for VA Pac OS/2 or WINDOWS NT, and the $PACDIR/save directory for VA Pac UNIX (PE.NEW, PE and PE-1 files).

UPDATE AND BACKUP: PACU15

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

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

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

.Permanent input-output files:
-Error messages
PAC7AE
-User parameters
PAC7AP

.Permanent input files:
-User parameter backup
PAC7EC (PE in directory SAVE)

.Transaction file:
-Update transactions
PAC7MC (MBPARMfile in directory INPUT)

.Output file
-User parameter backup
PAC7CE (PE.NEW in directory SAVE)

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

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

RECONSTRUCTION OF THE AE AND AP FILES: PACU80

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

.Permanent input files:
-User parameter backup
PAC7CE (PE.NEW in directory SAVE)
-Initial sequential image of
error messages
PAC7LE (AE0 in directory SAVE)

.Transaction file:
-Update transactions
PAC7MC (MBPARM file in directory INPUT)

.Permanent output files:
-Error messages to be rebuilt
PAC7AE
-User parameters to be recreated
PAC7AP

.Output report:
-Reconstruction report
PAC7IJ

.Sort file(s):
Not assigned

3.10.10. PARM: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) PARM BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        PARM PROCEDURE"
echo "                        ==============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : USER PARAMETER UPDATING
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AP.ini
PAC7MC=$PACINPUT'MBPARM'
export PAC7MC
. $PACDIR/assign/$1/PACSAVPE.ini
PAC7EC=$PACSAVPE
export PAC7EC
PAC7CE=$PACSAVPENEW
export PAC7CE
PAC7IJ=$PACTMP'PARMIJ.U15'
export PAC7IJ
PAC7DD=$PACTMP'PARMDD.U15'
export PAC7DD
echo "Execution : PACU15"
cobrun PACU15
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7AP.ini
 PAC7MC=$PACINPUT'MBPARM'
 export PAC7MC
. $PACDIR/assign/$1/PACSAVAE0.ini
 PAC7LE=$PACSAVAE0
 export PAC7LE
. $PACDIR/assign/$1/PACSAVPE.ini
 PAC7CE=$PACSAVPENEW
 export PAC7CE
 PAC7IJ=$PACTMP'PARMIJ.U80'
 export PAC7IJ
 echo "Execution : PACU80"
 cobrun PACU80
 RETURN=$?
 case $RETURN in
 0)
  echo "End of procedure"
  echo ""
  echo "Call the file PEBACKUP.ini"
  sh $PACDIR/assign/$1/PEBACKUP.ini
```

```
     ;;
    *)
     echo "Error in executing PACU80"
     ;;
    esac
    ;;
   8)
    echo "Error in executing PACU15"
    echo "Error 8 : Error on * input line"
    ;;
   4)
    echo "No reloading of files AE and AP"
    echo "End of procedure"
    echo ""
    echo "Call the file PEBACKUP.ini"
    sh $PACDIR/assign/$1/PEBACKUP.ini
    ;;
   *)
    echo "Error in executing PACU15"
    ;;
   esac
   if [ "$RETURN" != '0' -a "$RETURN" != '4' ]
   then
      sh $PACDIR/batch/proc/ERRPAUSE.ini
   fi
   exit $RETURN
```

4. VERSIONING UTILITIES

4.1. PEI: PRODUCTION ENVIRONMENT INTERFACE

4.1.1. PEI: OVERVIEW

PEI: INTRODUCTION

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

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

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

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

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

. The possibility to manually request a session freeze;

. Generation of purge requests for redundant frozen sessions;

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

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

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

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

PEI FILES

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

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

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

FILE SIZE

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

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

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

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

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

Each deletion is logical until a restoration procedure is performed.

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

4.1.2. INPE: FILE INITIALIZATION
4.1.2.1. INPE: INTRODUCTION

INPE: INTRODUCTION

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

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

EXECUTION CONDITION

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

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

ABENDS

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

USER INPUT

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

4.1.2.2. INPE: DESCRIPTION OF STEPS

INPE: DESCRIPTION OF STEPS

PEI INITIAL BACKUP: PACR01

```
.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error message file
  PAC7AE

.Input file:
 -User input file
  PAC7MB    (MBINPE file in directory INPUT)

.Output file:
 -PEI initial backup
  PAC7PP

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

.Sort file(s):
 Not assigned

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

4.1.2.3. INPE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) INPE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-------------------------------------------------------"
echo "                        INPE PROCEDURE"
echo "                        =============="
echo "Directory 'assign'               : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PEI - INITIALIZATION OF FILES
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PACSAVPP.ini
PAC7PP=$PACSAVPP
export PAC7PP
PAC7MB=$PACINPUT'MBINPE'
export PAC7MB
PAC7IB=$PACTMP'INPEIB.R01'
export PAC7IB
PAC7DD=$PACTMP'INPEDD.R01'
export PAC7DD
echo "Execution : PACR01"
cobrun PACR01
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 echo "-------------------------------------------------------"
 echo "*     Execute restoration procedure RSPE"
 echo "-------------------------------------------------------"
 ;;
8)
 echo "Error in executing PACR01"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PACR01"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

4.1.3. SVPE: FILE BACKUP
4.1.3.1. SVPE: INTRODUCTION

SVPE: INTRODUCTION

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

EXECUTION CONDITION

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

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

ABNORMAL EXECUTION

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

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

USER INPUT

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

4.1.3.2. SVPE: DESCRIPTION OF STEPS

SVPE: DESCRIPTION OF STEPS

PEI BACKUP: PACR60

.Permanent input files:
 -'Batch' PEI file
  PAC7AB
 -'On-line' PEI file
  PAC7AC
 -Data file
  PAC7AR
 -Error message file
  PAC7AE

.Output file:
 -PEI backup
  PAC7PP

.Input file:
 -Transaction file
  PAC7MB

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

.Return code(s):
 8: Unauthorized user.

4.1.3.3. SVPE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SVPE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------"
echo "                        SVPE PROCEDURE"
echo "                        =============="
echo "Directory 'assign'               : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PEI - FILES BACKUP
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# ************************************************************
. $PACDIR/assign/$1/PAC7AB.ini
. $PACDIR/assign/$1/PAC7AC.ini
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBSVPE'
export PAC7MB
. $PACDIR/assign/$1/PACSAVPP.ini
PAC7PP=$PACSAVPPNEW
export PAC7PP
PAC7IE=$PACTMP'SVPEIE.R60'
export PAC7IE
PAC7DD=$PACTMP'SVPEDD.R60'
export PAC7DD
echo "Execution : PACR60"
cobrun PACR60
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 echo ""
 echo "Call the file PPBACKUP.ini"
 sh $PACDIR/assign/$1/PPBACKUP.ini
 ;;
8)
 echo "Error in executing PACR60"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PACR60"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

4.1.4. RSPE: FILE RESTORATION
4.1.4.1. RSPE: INTRODUCTION

RSPE: PEI FILE RESTORATION

RSPE: INTRODUCTION

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

EXECUTION CONDITION

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

Batch procedure authorization option: Authorization level 4 is required.

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

ABNORMAL EXECUTION

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

USER INPUT

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

4.1.4.2. RSPE: DESCRIPTION OF STEPS

RSPE: DESCRIPTION OF STEPS

USER INPUT RECOGNITION: PTU004

.Input file:
CARTE

.Output file:
PAC7MB

.Permanent input file:
-Error message file
PAC7AE

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

.Return code(s):
-8: Unauthorized user

PEI RESTORATION: PACR61

.Input file:
-User input
PAC7MB

.Permanent input files:
-Error message file
PAC7AE
-Data file
PAC7AR
-PEI backup file
PAC7PP

.Permanent output files:
-'Batch' PEI file
PAC7AB
-'On-line' PEI file
PAC7AC

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

.Return code:
-8: Unauthorized user

4.1.4.3. RSPE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) RSPE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                         RSPE PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PEI - FILES RESTORATION
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
CARTE=$PACINPUT'MBRSPE'
export CARTE
PAC7MB=$PACTMP'MB'
export PAC7MB
PAC7DD=$PACTMP'RSPEDD.004'
export PAC7DD
echo "Execution: PTU004"
cobrun PTU004
RETURN=$?
case $RETURN in
0)
 ;;
8)
 echo "Error in executing PTU004"
 echo "Error 8: Unauthorized user"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PTU004"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AB.ini
. $PACDIR/assign/$1/PAC7AC.ini
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBRSPE'
export PAC7MB
. $PACDIR/assign/$1/PACSAVPP.ini
PAC7PP=$PACSAVPP
export PAC7PP
PAC7IF=$PACTMP'RSPEIF.R61'
```

```
export PAC7IF
PAC7DD=$PACTMP'RSPEDD.R61'
export PAC7DD
echo "Execution : PACR61"
cobrun PACR61
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
8)
 echo "Error in executing PACR61"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PACR61"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

4.1.5. PRPE: PRODUCTION ENVIRONMENT PRINTOUTS
4.1.5.1. PRPE: INTRODUCTION

PRPE: INTRODUCTION

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

EXECUTION CONDITION

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

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

ABENDS

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

4.1.5.2. PRPE: USER INPUT

PRPE: USER INPUT

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

Specific input:

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

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

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

4.1.5.3. PRPE: DESCRIPTION OF STEPS

PRPE: DESCRIPTION OF STEPS

PEI PRINTING: PACR10

.Permanent input files:
-'Batch' PEI file
PAC7AB
-'On-line' PEI file
PAC7AC
-Data file
PAC7AR
-Index file
PAC7AN
-Error-message file
PAC7AE

.Input transaction file:
-Printing requests
PAC7MB

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

.Sort file(s):
Not assigned

.Return code(s):
- 8: Unauthorized user

4.1.5.4. PRPE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) PRPE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         PRPE PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PEI - PRODUCTION ENVIRONMENT PRINTING
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# * .PRINTING REQUEST
# *  COL  2-3  : 'PL' (CARD CODE)
# *  COL  4    : '1' ENVIRONMENTS PER LIBRARY
# *  COL  5    : '1' LIBRARIES PER ENVIRONMENT
# *  COL  6    : '1' ENTITIES PER ENVIRONMENT
# *  COL  7    : '1' ENTITIES PER SESSION
# *  COL  8    : '1' ENVIRONMENTS PER VA Pac ENTITY
# *  COL  9    : '1' ENVIRONMENTS PER ENTITY (EXT. NAME)
# ************************************************************
. $PACDIR/assign/$1/PAC7AB.ini
. $PACDIR/assign/$1/PAC7AC.ini
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBPRPE'
export PAC7MB
PAC7IE=$PACTMP'PRPEIE.R10'
export PAC7IE
PAC7DD=$PACTMP'PRPEDD.R10'
export PAC7DD
echo "Execution : PACR10"
rtscgi PACR10
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
8)
 echo "Error in executing PACR10"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PACR10"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
```

```
    fi
    exit $RETURN
```

4.1.6. GRPE: TRANSACTION-GENERATION FOR REORGANIZATION
4.1.6.1. GRPE: INTRODUCTION

GRPE: INTRODUCTION

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

PRINT

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

EXECUTION CONDITION

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

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

ABENDS

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

USER INPUT

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

4.1.6.2. GRPE: DESCRIPTION OF STEPS

GRPE: DESCRIPTION OF STEPS

GENERATION OF TRANSACTIONS FOR REORGANIZATION: PACR40

```
.Permanent input files:
 -'Batch' PEI file
  PAC7AB
 -'On-line' PEI file
  PAC7AC
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error message file
  PAC7AE

.Input file:
 -User input
  PAC7MB

.Output file:
 -Generated trans. for reorganization
  PAC7MV

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

.Sort file(s):
 Not assigned

.Return code(s):
 -8:  Unauthorized user
```

4.1.6.3. GRPE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GRPE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-------------------------------------------------------"
echo "                        GRPE PROCEDURE"
echo "                        =============="
echo "Directory 'assign'               : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PEI - TRANSACTION GENERATION FOR REORGANIZATION
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# ************************************************************
. $PACDIR/assign/$1/PAC7AB.ini
. $PACDIR/assign/$1/PAC7AC.ini
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBGRPE'
export PAC7MB
PAC7MV=$PACINPUT'MVGRPE'
export PAC7MV
PAC7IK=$PACTMP'GRPEIK.R40'
export PAC7IK
PAC7DD=$PACTMP'GRPEDD.R40'
export PAC7DD
echo "Execution : PACR40"
cobrun PACR40
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
8)
 echo "Error in executing PACR40"
 echo "Error 8 : Error on * input line"
 ;;
4)
 echo "End of procedure"
 echo "No purge transaction"
 ;;
*)
 echo "Error in executing PACR40"
 ;;
esac
if [ "$RETURN" != '0' -a "$RETURN" != '4' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

4.1.7. HIPE: AUTOMATIC SESSION FREEZE
4.1.7.1. HIPE: INTRODUCTION

HIPE: INTRODUCTION

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

EXECUTION CONDITION

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

ABENDS

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

4.1.7.2. HIPE: USER INPUT

HIPE: USER INPUT

A required '*' line:

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

An optional session freeze line:

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

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

This line may be entered in order to:

.Give a specific freeze comment,

.Force the session number.

PRINTED REPORTS

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

4.1.7.3. HIPE: DESCRIPTION OF STEPS

HIPE: DESCRIPTION OF STEPS

DATABASE CONSISTENCY CHECK: PTUBAS

```
.Permanent input files:
 -Data file
  PAC7AR
 -Error message file
  PAC7AE
 -Update serialization file
  PAC7LO

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

.Return code(s):
 -0:  OK.
 -4:  Database invalid, STOP triggered.
```

AUTOMATIC SESSION FREEZE: PACR30

```
.Permanent input files:
 -'Batch' PEI file
  PAC7AB
 -'On-line' PEI file
  PAC7AC
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Journal file
  PAC7AJ
 -Error message file
  PAC7AE


.Input transaction file:
 -Session freeze requests
  PAC7MB

.Output report:
 -Execution report
  PAC7IG

.Work files:
  PAC7MW
  PAC7WB
```

4.1.7.4. HIPE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) HIPE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                         HIPE PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PAC7AJ.ini
echo "Directory of the AJ file          : »dirname $PAC7AJ.»"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PEI - AUTOMATIC DATABASE SESSION FREEZE
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE '*' IDENTIFICATION LINE (REQUIRED)
# *  COL  2    : '*' LINE CODE
# *  COL  3-10 : USER CODE (UUUUUUUU)
# *  COL 11-18 : USER PASSWORD (PPPPPPPPP)
# *  COL 19-21 : '***' INTER-LIBRARY (REQUIRED)
# * .FREEZE LINE (OPTIONAL)
# *  COL  2-3  : 'X1'   LINE CODE
# *  COL  4-7  : 'HIST' FREEZE REQUEST
# *  COL  8-67 : FREEZE COMMENTS
# *  COL 68-71 : FORCING THE NUMBER OF SESSION TO BE FROZEN
# *            : THIS NUMBER MUST BE GREATER THAN THE CURRENT
# *            : SESSION NUMBER (SSSS).
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7DS=$PACTMP'HIPEDS.BAS'
export PAC7DS
. $PACDIR/assign/$1/SEMLOCK.ini
echo "Execution : PTUBAS"
rtscgi PTUBAS
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AB.ini
 . $PACDIR/assign/$1/PAC7AC.ini
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7AJ.ini
 . $PACDIR/assign/$1/PAC7AN.ini
 . $PACDIR/assign/$1/PAC7AR.ini
 PAC7MB=$PACINPUT'MBHIPE'
 export PAC7MB
 PAC7MW=$PACTMP'MW'
 export PAC7MW
 PAC7WB=$PACTMP'WB'
 export PAC7WB
 PAC7IG=$PACTMP'HIPEIG.R30'
 export PAC7IG
 echo "Execution : PACR30"
```

```
 cobrun PACR30
 RETURN=$?
 case $RETURN in
 0)
  echo "End of procedure"
  echo ""
  echo "Deletion of the temporary files"
  rm -f $PACTMP'MW'
  rm -f $PACTMP'WB'
  ;;
 8)
  echo "Error in executing PACR30"
  echo "Error 8 : Error on * input line"
  ;;
 *)
  echo "Error in executing PACR30"
  ;;
 esac
 ;;
4)
 echo "Error in executing PTUBAS"
 echo "Database unavailable"
 ;;
*)
 echo "Error in executing PTUBAS"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

4.1.8. SIPE: PRODUCTION TURNOVER SIMULATION
4.1.8.1. SIPE: INTRODUCTION

SIPE: INTRODUCTION

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

Three SIPE operations are available:

1. Simulation of update with GPRT:

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

2. Simulation of environment transfer:

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

3. Existing systems retrieval:

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

EXECUTION CONDITION

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

Batch procedure access authorization: Level 3 is required.

ABENDS

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

4.1.8.2. SIPE: USER INPUT

SIPE: USER INPUT

A required '*' line.

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

An 'EE' line identifying the environment is required.

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

An 'EU' line for each entity to update:

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

4.1.8.3. SIPE: DESCRIPTION OF STEPS

SIPE: DESCRIPTION OF STEPS

PRODUCTION TURNOVER: PACR22

.Permanent input files:
-'Batch' PEI file
PAC7AB
-'On-line' PEI file
PAC7AC
-Data file
PAC7AR
-Index file
PAC7AN
-Error message file
PAC7AE

.Transaction file:
-User input
PAC7MB

.Output file:
-Transactions used to build data cards
for TRANSFER utilities
PAC7MT

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

4.1.8.4. SIPE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SIPE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        SIPE PROCEDURE"
echo "                        =============="
echo "Directory 'assign'               : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PEI - PRODUCTION TURNOVER SIMULATION
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE '*' LINE                          (REQUIRED)
# *  COL  2    : '*'  CARD CODE
# *  COL  3-10 : USER CODE
# *  COL 11-18 : PASSWORD
# *  COL 19-21 : LIBRARY CODE
# *  COL 22-25 : SESSION NUMBER
# *            : (BLANK IF CURRENT)
# *  COL 26    : SESSION STATUS (' ' ou 'T')
# *  COL 61-66 : GENERATION DATE (YYMMDD), IF SESSION
# *            : IS NOT CURRENT SESSION
# * .ONE LINE IDENTIFYING THE ENVIRONMENT   (REQUIRED)
# *  COL  2-3  : 'EE' CARD CODE
# *  COL  4    : ENTITY TYPE
# *  COL  5    : TARGET ENVIRONMENT TYPE
# *  COL  6    : SOURCE ENVIRONMENT TYPE
# * .ONE ENTITY IDENTIFICATION LINE PER ENTITY TO UPDATE
# *  COL  2-3  : 'EU' CARD CODE
# *  COL  4-11 : VA Pac ENTITY CODE
# *  COL 12-19 : ENTITY EXTERNAL NAME IN TARGET ENVIRONMENT
# *            : (IF DIFFERENT FROM VA Pac CODE)
# *  COL 20-27 : ENTITY EXTERNAL NAME IN SOURCE ENVIRONMENT
# *            : (IF TRANSFER WITH RENAME)
# ************************************************************
. $PACDIR/assign/$1/PAC7AB.ini
. $PACDIR/assign/$1/PAC7AC.ini
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBSIPE'
export PAC7MB
PAC7MT=$PACINPUT'MVSIPE'
export PAC7MT
PAC7IE=$PACTMP'SIPEIE.R22'
export PAC7IE
PAC7DD=$PACTMP'SIPEDD.R22'
export PAC7DD
echo "Execution : PACR22"
cobrun PACR22
RETURN=$?
case $RETURN in
```

```
    0)
     echo "End of procedure"
     ;;
    8)
     echo "Error in executing PACR22"
     echo "Error 8 : Error on * input line"
     ;;
    *)
     echo "Error in executing PACR22"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

4.2. PAC/TRANSFER

PAC/TRANSFER: INTRODUCTION

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

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

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

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

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

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

FUNCTIONALITIES

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

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

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

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

OPERATING MODE

1. UPDATING THE TRANSFER PARAMETERS

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

2. COMPRESSING THE ARCHIVED JOURNAL

Optional process (depending on the site).

3. CREATING THE TRANSFER FILE

4. PREPARING THE DSMS ENVIRONMENT

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

5. GENERATING THE TRANSFER TRANSACTIONS

6. UPDATING THE VISUALAGE PACBASE DATABASE

7. REINITIALIZING THE DSMS ENVIRONMENT

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

4.2.1. TRUP: TRANSFER-PARAMETER UPDATE
4.2.1.1. TRUP: INTRODUCTION

TRUP: INTRODUCTION

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

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

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

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

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

TRANSFER PARAMETERS

1.1. SESSION:

Specify one source session and at least one target session.

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

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

1.2. LIBRARY:

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

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

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

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

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

1.3. USER:

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

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

1.4. DSMS CHANGE NUMBER:

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

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

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

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

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

EXECUTION CONDITION

None.

PRINTED REPORT

Printout of the parameter-file contents.

4.2.1.2. TRUP: USER INPUT

TRUP: USER INPUT

. User identification line (required)

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

. Session-selection line

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

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

. Library-selection line

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

. User-selection line

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

. DSMS-change selection line

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

. Multiple-deletion request line

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

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

EXAMPLES:

EXAMPLE 1

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

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

EXAMPLE 2

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

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

EXAMPLE 3

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

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

EXAMPLE 4

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

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

EXAMPLE 5

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

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

4.2.1.3. TRUP: DESCRIPTION OF STEPS

TRUP: DESCRIPTION OF STEPS

UPDATE OF THE SELECTION PARAMETERS: PTUG10

This step updates the selection-parameter file.

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE

.Transaction file
 -User input
  PAC7MA

.Output file:
 -List of Transfer Sets
  PAC7ML

.Input/output file:
 -Parameter file
  PAC7UV

.Work file:
 -Transaction file with generated multiple deletions
  PAC7MV

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

SELECTION-PARAMETER PRINTOUT: PTUG11

.Permanent input files:
 -Data file
  PAC7AR
 -Error-message file
  PAC7AE
 -Parameter file
  PAC7UV

.Output file:
 -List of target sessions
  PAC7GL

.Output report:
 -Printout of parameter table
  PAC7ET

PRINTING OF TARGET-SESSION LIST: PTUG12

.Input files:
 -Data file
  PAC7AR
 -Parameter file
  PAC7UV
 -Error-message file
  PAC7AE
 -Target-session list
  PAC7GL
 -List of Sets
  PAC7ML

.Sort file(s):
  Not assigned

.Output report:
 -Target-session list printout
  PAC7ET

4.2.1.4. TRUP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) TRUP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         TRUP PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TRANSFER - UPDATE OF THE PARAMETERS FILE
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7UV.ini
PAC7MA=$PACINPUT'MBTRUP'
export PAC7MA
PAC7ML=$PACTMP'ML'
export PAC7ML
PAC7MV=$PACINPUT'MVTRUP'
export PAC7MV
PAC7DD=$PACTMP'TRUPDD.G10'
export PAC7DD
PAC7ET=$PACTMP'TRUPET.G10'
export PAC7ET
echo "Execution : PTUG10"
cobrun PTUG10
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG10"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7UV.ini
PAC7GL=$PACTMP'GL'
export PAC7GL
PAC7ET=$PACTMP'TRUPET.G11'
export PAC7ET
echo "Execution : PTUG11"
cobrun PTUG11
RETURN=$?
case $RETURN in
0)
 ;;
```

```
    *)
     echo "Error in executing PTUG11"
     echo "Error $RETURN"
     sh $PACDIR/batch/proc/ERRPAUSE.ini
     exit $RETURN
     ;;
    esac
    # ************************************************************
    . $PACDIR/assign/$1/PAC7AE.ini
    . $PACDIR/assign/$1/PAC7AR.ini
    . $PACDIR/assign/$1/PAC7UV.ini
    PAC7GL=$PACTMP'GL'
    export PAC7GL
    PAC7ML=$PACTMP'ML'
    export PAC7ML
    PAC7ET=$PACTMP'TRUPET.G12'
    export PAC7ET
    echo "Execution : PTUG12"
    cobrun PTUG12
    RETURN=$?
    case $RETURN in
    0)
     ;;
    *)
     echo "Error in executing PTUG12"
     echo "Error $RETURN"
     sh $PACDIR/batch/proc/ERRPAUSE.ini
     exit $RETURN
     ;;
    esac
    # ************************************************************
    echo "End of procedure"
    echo ""
    echo "The output file MVTRUP will be processed by UPDT"
    echo "(created in the directory $PACINPUT)"
    echo ""
    echo "Deletion of the temporary files"
    rm -f  $PACTMP'GL'
    rm -f  $PACTMP'ML'
    exit $RETURN
```

4.2.2. TRJC: COMPRESSION OF ARCHIVED JOURNAL
4.2.2.1. TRJC: INTRODUCTION

TRJC: INTRODUCTION

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

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

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

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

NOTES:

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

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

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

EXECUTION CONDITION

None.

RESULT

A smaller archived Journal including effective transactions only.

OUTPUT REPORT

Statistical data on the TRJC execution.

4.2.2.2. TRJC: USER INPUT

TRJC: USER INPUT

. User identification line (required)

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

. Options

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

4.2.2.3. TRJC: DESCRIPTION OF STEPS

TRJC: DESCRIPTION OF STEPS

COMPRESSION (FIRST STAGE): PTUG05

.Permanent input files:
 -Sequential journal
  PAC7PJ
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE

.Transaction file:
 -User input
  PAC7MB

.Output file:
 -Temporary journal
  PAC7GP

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

.Sort file(s):
  Not assigned

COMPRESSION (SECOND STAGE): PTUG06

.Input transaction file:
 -Temporary file
  PAC7GP

.Output file:
 -Sequential compressed file
  PAC7PK

.Sort file(s):
  Not assigned

CLASSIFICATION OF DELETIONS/CREATIONS: PTUG07

.Input file:
 -Index file
   PAC7AN

.Input transaction files:
 -Temporary journal
   PAC7PK

.Output file:
 -Compressed sequential file
   PAC7PL

.Sort file(s):
  Not assigned

4.2.2.4. TRJC: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) TRJC BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         TRJC PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TRANSFER - COMPRESS OF THE JOURNAL TRANSACTIONS
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PACSAVPJ.ini
PAC7PJ=$PACSAVPJ
export PAC7PJ
PAC7MB=$PACINPUT'MBTRJC'
export PAC7MB
PAC7GP=$PACTMP'GP'
export PAC7GP
PAC7DD=$PACTMP'TRJCDD.G05'
export PAC7DD
PAC7ET=$PACTMP'TRJCET.G05'
export PAC7ET
echo "Execution : PTUG05"
cobrun PTUG05
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG05"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
PAC7PK=$PACTMP'PK'
export PAC7PK
PAC7GP=$PACTMP'GP'
export PAC7GP
echo "Execution : PTUG06"
cobrun PTUG06
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG06"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
```

```
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PACSAVJT.ini
PAC7PL=$PACSAVJT
export PAC7PL
PAC7PK=$PACTMP'PK'
export PAC7PK
echo "Execution : PTUG07"
cobrun PTUG07
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG07"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
echo "Deletion of the temporary files"
rm -f  $PACTMP'GP'
rm -f  $PACTMP'PK'
exit $RETURN
```

4.2.3. TRPF: TRANSFER-FILE CREATION
4.2.3.1. TRPF: INTRODUCTION

TRPF: INTRODUCTION

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

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

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

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

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

EXECUTION CONDITIONS

None.

RESULT

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

4.2.3.2. TRPF: USER INPUT

TRPF: USER INPUT

. User identification line (required)

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

. Transaction Set for processing selection line (required)

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

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

4.2.3.3. TRPF: DESCRIPTION OF STEPS

TRPF: DESCRIPTION OF STEPS

CREATION OF TRANSFER FILE: PTUG50

.Permanent input files:
 -Index file
  PAC7AR
 -Error-message file
  PAC7AE
 -Parameter file
  PAC7UV
 -Sequential or compressed file
  PAC7JT

.Transaction file:
 -User input
  PAC7MB

.Output files:
 -Sequential transfer journal
  PAC7TJ

.Sort file(s):
  Not assigned

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

4.2.3.4. TRPF: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) TRPF BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         TRPF PROCEDURE"
echo "                         ==============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TRANSFER - GENERATION OF THE TRANSFER TRANSACTIONS
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7UV.ini
. $PACDIR/assign/$1/PACSAVJT.ini
PAC7JT=$PACSAVJT
# If TRJC has not been executed
# . $PACDIR/assign/$1/PACSAVPJ.ini
# PAC7JT=$PACSAVPJ
export PAC7JT
. $PACDIR/assign/$1/PACSAVTJ.ini
PAC7TJ=$PACSAVTJ
export PAC7TJ
PAC7MB=$PACINPUT'MBTRPF'
export PAC7MB
PAC7DD=$PACTMP'TRPFDD.G50'
export PAC7DD
PAC7ER=$PACTMP'TRPFER.G50'
export PAC7ER
PAC7ET=$PACTMP'TRPFET.G50'
export PAC7ET
echo "Execution : PTUG50"
cobrun PTUG50
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG50"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
exit $RETURN
```

4.2.4. TRDU: DSMS-ENVIRONMENT PREPARATION
4.2.4.1. TRDU: INTRODUCTION

TRDU: INTRODUCTION

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

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

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

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

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

TRDU takes into account the following additional parameters:

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

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

The TRDU procedure produces two files:

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

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

This update must be executed BEFORE the transfer operation.

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

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

EXECUTION CONDITION

None.

RESULT

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

4.2.4.2. TRDU: USER INPUT

TRDU: USER INPUT

. User identification line (required)

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

. TRANSACTION SET selection line (required)

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

One and only one LT-type line is required.

4.2.4.3. TRDU: DESCRIPTION OF STEPS

TRDU: DESCRIPTION OF STEPS

SELECTION OF SETS: PTUG42

.Input files:
 -Data file
  PAC7AR
 -Error-messages file
  PAC7AE
 -Parameter file
  PAC7UV
 -User input
  PAC7MB

.Output file:
 -SETS file
  PAC7BM

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

PREPARATION OF DSMS BEFORE TRANSFERS: PTUG44

.Input files:
 -Parameter file
  PAC7UV
 -Error-message file
  PAC7AE
 -Data file
  PAC7AR
 -VisualAge Pacbase element file
  PACDDC
 -Batch-transaction file
  PAC7MB

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

.Output report:
 -Execution report
  PAC7ET

GENERATION OF TARGET CHANGE TRANSACTIONS: PTUG46

.Input files:
 -Error-message file
  PAC7AE
 -Data file
  PAC7AR
 -Preparation file for target-Change authorizations
  PAC7GC

.Output files:
 -Target before-transfer creation transactions

PAC7CC
-Target after-transfer deletion transactions
PAC7SC

.Sort file:
Not assigned

.Output report:
-Execution report
PAC7ET

4.2.4.4. TRDU: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) TRDU BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                         TRDU PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TRANSFER - GENERATION OF THE DSMS TRANSACTIONS
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7UV.ini
PAC7MB=$PACINPUT'MBTRDU'
export PAC7MB
PAC7BM=$PACTMP'BM'
export PAC7BM
PAC7DD=$PACTMP'TRDUDD.G42'
export PAC7DD
PAC7ET=$PACTMP'TRDUET.G42'
export PAC7ET
echo "Execution : PTUG42"
cobrun PTUG42
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG42"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7UV.ini
. $PACDIR/assign/$1/PAC7DC.ini
PAC7CI=$PACTMP'CI'
export PAC7CI
PAC7SI=$PACTMP'SI'
export PAC7SI
PAC7GC=$PACTMP'GC'
export PAC7GC
PAC7ET=$PACTMP'TRDUET.G44'
export PAC7ET
echo "Execution : PTUG44"
cobrun PTUG44
RETURN=$?
case $RETURN in
```

```
0)
 ;;
*)
 echo "Error in executing PTUG44"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7UV.ini
PAC7CC=$PACTMP'CC'
export PAC7CC
PAC7SC=$PACTMP'SC'
export PAC7SC
PAC7GC=$PACTMP'GC'
export PAC7GC
PAC7ET=$PACTMP'TRDUET.G46'
export PAC7ET
echo "Execution : PTUG46"
cobrun PTUG46
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG46"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
cat $PACTMP'CC' $PACTMP'SI' > $PACINPUT'MVDUAV'
cat $PACTMP'CI' $PACTMP'SC' > $PACINPUT'MVDUAP'
echo "The output files MVDUAV and MVDUAP will be processed by"
echo "DUPT (created in the directory $PACINPUT)"
echo ""
echo "Deletion of the temporary files"
rm -f  $PACTMP'CC'
rm -f  $PACTMP'CI'
rm -f  $PACTMP'SI'
rm -f  $PACTMP'SC'
exit $RETURN
```

4.2.5. UPDATE OF DSMS FUNCTION BEFORE VA PAC UPDATE

UPDATE OF DSMS BEFORE VA PAC UPDATE

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

4.2.6. TRRP: GENERATION OF TRANSFER TRANSACTIONS
4.2.6.1. TRRP: INTRODUCTION

TRRP: INTRODUCTION

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

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

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

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

Each combination of criteria corresponds to a TRRP execution mode.

1 Standard execution mode (by Transaction Set):

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

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

However, you must run as many TRRP executions as there are target sessions:

A specific attribute -- SESSION PROCESSED -- is automatically positioned in the Parameter file once all transactions have been generated for a given session.

As a result, if this attribute is positioned for a given session (see also the other execution modes, described in Paragraphs 2 and 3), transactions for that session will not be generated and TRRP will automatically proceed with the next target session, as listed in the Parameter file.

This execution mode brings an automatic control over your transfer operations since it avoids duplicating transactions which could otherwise happen when prior TRRP executions have been run.

The TRRP standard execution mode is therefore recommended for sites where Pac/transfer operations involve large volumes of transactions.

A Warning message will tell you when all sessions have been dealt with.

Generated transactions must then be used by the VisualAge Pacbase batch update procedure (UPDT).

You may prefer to concatenate all TRRP subsequent outputs and run the UPDT procedure only once.

2. Execution mode by Session:

Transaction Set code different from '*****'
Target session: 'nnnnT' or '*****'

TRRP considers transactions that belong to the selected Transaction Set only.

1. If you have selected a target session, transactions are generated for this session only.

2. If you have selected all sessions ('*****'), transactions are systematically generated for all target sessions, all in one TRRP execution.

>>>> A specific attribute -- SESSION PROCESSED -- is automatically positioned in the Parameters file once all transactions have been generated for a given session.

Generated transactions must then be used by the VA Pac batch update procedure (UPDT).

3. Execution mode for all Sets and all target sessions:

. Transaction Set code: '*****'
. Target session number: '****'

Transactions are systematically generated for all Sets and for all their respective target sessions.

>>>> A specific attribute -- SESSION PROCESSED -- is automatically positioned in the Parameters file once all transactions have been generated for a given session.

Generated transactions must then be used by the VA Pac batch update procedure (UPDT).

EXECUTION CONDITIONS

The Transfer file must exist (created by the TRPF procedure). Authorization level 4 is required to run a TRRP execution.

RESULT OBTAINED

Transfer transactions formatted for the VA Pac UPDT batch update procedure.

4.2.6.2. TRRP: USER INPUT

TRRP: USER INPUT

. User identification line (required)

Pos.	Len.	Value	Significance
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	5		Selection of target session(s):
		blank	. All target sessions (default), one session processed per TRRP execution. This value cannot be used when all Transaction sets are selected
		nnnnT	. Target session number (required)
		'*****'	. All target sessions processed in one TRRP execution
40	5		Selection of Transaction Set(s):
		lllll	Transaction Set code
		'*****'	All Transaction Sets

. Sort Options line

Pos.	Len.	Value	Significance
2	1	'J'	Line code
4	1	' '	Chronological list
		'N'	No chronological list
5	1	' '	List by user
		'N'	No list by user
6	1	' '	List by library
		'N'	No list by library

4.2.6.3. TRRP: DESCRIPTION OF STEPS

TRRP: DESCRIPTION OF STEPS

PREPARATION OF EXTRACTION: PTUG60

.Permanent input files:
 -Index file
  PAC7AR
 -Error messages
  PAC7AE
 -Parameter-setting file
  PAC7UV
 -Compressed journal file
  PAC7JT

.Transaction file:
 -User input
  PAC7MB

.Output file:
 -Parameter-line file
  PAC7BM
 -Temporary journal file
  PAC7PJ

.Output reports:
 -Transfer statistics
  PAC7ET
 -User check
  PAC7DD

.Return code(s):
 - 4: If there are no more sessions to be extracted

EXTRACTION: PACX

This step extracts transactions based on user input.

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE
 -Transactions selected on Journal
  PAC7PJ

.Input transaction file:
 -User input
  PAC7MB

.Work files
 -User input
  PAC7BM
 -Journal transactions (EXPJ)
  PAC7MJ
 -Extracted transactions
  PAC7WD

.Output file:
 -Transactions extracted for UPDT
  PAC7MV

.Sort file(s):
  Not assigned

.Output reports:
 -General program-stream printout
  PAC7IA
 -List of errors on input transactions
  PAC7DD
 -Extraction list report(s)
  PAC7EE
  PAC7EP
  PAC7EQ
  PAC7EZ

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

POSITIONNING THE 'PROCESSED SESSION' ATTRIBUTE: PTUG61

.Permanent input files:
 -Index file
  PAC7AR
 -Error-message file
  PAC7AE

.Input transaction file
 -User input
  PAC7MB

.Input/Output file:
 -Parameter-settings
  PAC7UV

.Output report(s):
 -Transfer statistics
  PAC7ET

4.2.6.4. TRRP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) TRRP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         TRRP PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TRANSFER - GENERATION OF THE UPDT TRANSACTIONS
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7UV.ini
. $PACDIR/assign/$1/PACSAVTJ.ini
PAC7JT=$PACSAVTJ
export PAC7JT
PAC7MB=$PACINPUT'MBTRRP'
export PAC7MB
PAC7BM=$PACTMP'MB'
export PAC7BM
PAC7PJ=$PACTMP'PJ'
export PAC7PJ
PAC7DD=$PACTMP'TRRPDD.G60'
export PAC7DD
PAC7ET=$PACTMP'TRRPET.G60'
export PAC7ET
echo "Execution : PTUG60"
cobrun PTUG60
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG60"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7UV.ini
PAC7PJ=$PACTMP'PJ'
export PAC7PJ
PAC7MB=$PACTMP'MB'
export PAC7MB
PAC7BM=$PACTMP'BM'
export PAC7BM
PAC7MJ=$PACTMP'MJ'
```

```
export PAC7MJ
PAC7WD=$PACTMP'WD'
export PAC7WD
PAC7MV=$PACINPUT'MVTRRP'
export PAC7MV
PAC7IA=$PACTMP'TRRPIA.PAC'
export PAC7IA
PAC7DD=$PACTMP'TRRPDD.PAC'
export PAC7DD
PAC7EE=$PACTMP'TRRPEE.PAC'
export PAC7EE
PAC7EP=$PACTMP'TRRPEP.PAC'
export PAC7EP
PAC7EQ=$PACTMP'TRRPEQ.PAC'
export PAC7EQ
PAC7EZ=$PACTMP'TRRPEZ.PAC'
export PAC7EZ
echo "Execution : PACX"
cobrun PACX
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PACX"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
8)
 echo "Error in executing PACX"
 echo "Error 8: Unauthorized user"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
4)
 echo "Error in executing PACX"
 echo "No list selection required"
 echo "End of procedure"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7UV.ini
PAC7MB=$PACTMP'MB'
export PAC7MB
PAC7ET=$PACTMP'TRRPET.G61'
export PAC7ET
echo "Execution : PTUG61"
cobrun PTUG61
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUG61"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
echo "The output file MVTRRP will be processed by UPDT"
echo "(created in the directory $PACINPUT)"
echo ""
echo "Deletion of the temporary files"
rm -f  $PAC7BM
rm -f  $PAC7MB
```

```
    rm -f  $PAC7MJ
    rm -f  $PAC7PJ
    rm -f  $PAC7WD
    exit $RETURN
```

4.2.7. UPDATE OF THE VISUALAGE PACBASE DATABASE

UPDATE OF THE VISUALAGE PACBASE DATABASE

The VisualAge Pacbase Database is updated via the UPDT procedure, taking the Transfer file -- created by the TRRP procedure -- as input.

In the case of a 'standard processing' of the generation of transfer transactions (see previous subchapter), the following procedures may be executed several times:

. TRRP (Generation of transfer transactions),

. UPDT (Update of the VA Pac Database).

4.2.8. REINITIALIZATION OF THE DSMS ENVIRONMENT

REINITIALIZATION OF THE DSMS ENVIRONMENT

This procedure resets update authorizations on the selected source and target Changes as they were before the transfer operation.

This initial state is obtained by running the DSMS update procedure (DUPT), using as input transactions the contents of the file resulting from the DSMS Environment Preparation procedure (TRDU).

4.3. TEAMCONNECTION

INTRODUCTION

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

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

The Interface enables the user to know:

- In VA Pac, the TeamConnection 'target' contexts where the generated objects are managed, i.e.: Family, Component, Release, and WorkArea;

- In TeamConnection, the 'source' context of these objects in VA Pac: Library Code, Session Number, User Code, Generation Date and Time. The 'source' context is defined in the 'Configurable Fields' specific to the Interface.

The Interface performs two main actions:

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

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

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

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

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

4.3.1. TCGP: PREPARATION FOR IMPORT INTO TEAMCONNECTION
4.3.1.1. TCGP: INTRODUCTION

TCGP: INTRODUCTION

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

CONDITION D'EXECUTION

The GPRT must be run first.

4.3.1.2. TCGP: DESCRIPTION OF STEPS

PREPARATION OF IMPORT ACTIONS FOR TEAMC: PTC100

.Permanent input files:
 -Target LIBRARY and SESSION file
  PAC7TS
 -Error messages
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN

.Input file:
 -VA Pac generated code:
  PAC7JB

.Work file:
 -PAF standard KSDS file
  SYSPAF

.Output report:
 -Execution-related errors
  PAC7ET

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

4.3.1.3. TCGP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) TCGP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                         PROCEDURE TCGP"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TEAM CONNECTION - PREPARING IMPORT
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7TS.ini
PAC7ET=$PACTMP'TCGPET.100'
export PAC7ET
PAC7JB=$PACINPUT'MBTCGP'
export PAC7JB
PAC7BJ=$PACINPUT'MVTCGP'
export PAC7BJ
SYSPAF=$PACTMP'TCGPSY'
export SYSPAF
echo "Execution : PTC100"
cobrun PTC100
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTC100"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
echo "Deletion of the temporary files"
rm -f $SYSPAF*
exit $RETURN
```

4.3.2. TCCI: INTER-ENVIRONMENT INTEGRITY CHECK
4.3.2.1. TRCI: INTRODUCTION

TCCI: INTRODUCTION

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

EXECUTION CONDITION

The interface must not be currently open.

4.3.2.2. TRCI: DESCRIPTION OF STEPS

EXTRACTION OF OCCURRENCES WITH TYPE CODE $7B: PTC400

.Permanent input/output files:
 -LIBRARY and SESSION file
  PAC7TS
 -Error messages
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN

.Input file:
 -Manager's identification
  PAC7CA

.Work file:
 -PAF standard KSDS file
  SYSPAF

.Output report:
 -Detected errors
  PAC7ET

.Output file:
 -Extracted transactions
  PAC7RT

INTER-ENVIRONMENT CHECK: PTC440

.Permanent input files:
 -LIBRARY and SESSION file
  PAC7TS
 -Error messages
  PAC7AE

.Input files:
 -Manager's identification
  PAC7CA
 -List of generated objects found in TeamConnection
  PAC7UN
 -List of generated objects stored in VA Pac
  PAC7UM

.Output report:
 -Detected errors
  PAC7ET

.Output file:
 -'Correction'-transaction file
   PAC7UR

GENERATION OF BATCH UPDATE TRANSACTIONS: PTC220

.Permanent input files:
 -Error messages
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN

.Input file:
 -'Correction' transactions
  PAC7UR

.Work file:

-PAF standard KSDS file
SYSPAF

.Output report:
-Check report
PAC7ET

.Output file:
-VA-Pacbase update transactions
PAC7MV

4.3.2.3. TCCI: EXECUTION

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) TCCI BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         TCCI PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TEAM CONNECTION - INTEGRITY CHECKING
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7TS.ini
PAC7CA=$PACINPUT'MBTCCI'
export PAC7CA
PAC7ET=$PACTMP'TCCIET.400'
export PAC7ET
PAC7RT=$PACTMP'RT'
export PAC7RT
SYSPAF=$PACTMP'TCCISY'
export SYSPAF
echo "Execution : PTC400"
cobrun PTC400
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTC400"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7TS.ini
PAC7CA=$PACINPUT'MBTCCI'
export PAC7CA
PAC7UM=$PACTMP'RT'
export PAC7UM
PAC7UN=$PACTMP'UN'
export PAC7UN
PAC7UR=$PACTMP'UR'
export PAC7UR
PAC7ET=$PACTMP'TCCIET.440'
export PAC7ET
echo "Execution : PTC440"
cobrun PTC440
```

```
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTC440"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7MV=$PACINPUT'MVTCCI'
export PAC7MV
SYSPAF=$PACTMP'TCCISY'
export SYSPAF
PAC7UR=$PACTMP'UR'
export PAC7UR
PAC7ET=$PACTMP'TCCIET.220'
export PAC7ET
echo "Execution : PTC220"
cobrun PTC220
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTC220"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
echo "Deletion of the temporary files"
rm -f  $PACTMP'RT'
rm -f  $PACTMP'UN'
rm -f  $PACTMP'UR'
rm -f  $SYSPAF*
exit $RETURN
```

4.3.3. TCLS: LIBRARY-SESSION UPDATE
4.3.3.1. TCLS: INTRODUCTION

TCLS: INTRODUCTION

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

EXECUTION CONDITION

None.

4.3.3.2. TCLS: USER INPUT

TCLS : ENTREES UTILISATEUR

USER INPUT

One line for each update request.

Parameter line for target session update:

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

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

Parameter line for update of target libraries:

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

PRINTED RESULT

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

4.3.3.3. TCLS: DESCRIPTION OF STEPS

LIBRARY- AND SESSION- FILE UPDATE: PTC010

.Permanent input/output file:
-TARGET LIBRARY and SESSION file
PAC7TS

.Permanent input file:
-Error messages
PAC7AE

.Input transaction file:
-User input
PAC7MV

.Output report:
-Update report
PAC7ET

PRINTING OF LIBRARY AND SESSION FILE: PTC030

.Permanent input files:
-TARGET LIBRARIES and SESSIONS
PAC7TS
-Error messages
PAC7AE

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

4.3.3.4. TCLS: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) TCLS BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         PROCEDURE TCLS"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : TEAM CONNECTION - LIBRARY AND SESSION UPDATE
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7TS.ini
PAC7MV=$PACINPUT'MBTCLS'
export PAC7MV
PAC7ET=$PACTMP'TCLSET.010'
export PAC7ET
echo "Execution : PTC010"
cobrun PTC010
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTC010"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7TS.ini
PAC7ET=$PACTMP'TCLSET.030'
export PAC7ET
echo "Execution : PTC030"
cobrun PTC030
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTC030"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
exit $RETURN
```

5. MANAGER'S UTILITIES

5.1. SESSION MANAGEMENT

5.1.1. ESES - CSES: INTRODUCTION

ESES - CSES: INTRODUCTION

The VA Pac session number cannot be greater than 9999.

When the session number is close to 9999, the utility program re-assigns all the session numbers, by incrementing the numbers of frozen sessions by 1 (starting from session 0001 or from a session chosen by the Administrator).

NOTE: The freeze is performed by the UPDT procedure. It increments the current session number.

This reassignment is carried out on sequential images of the files that include the session number, i.e. the backup files of the Database (PC), of the Journal (PJ), of the Print-Generation requests (PG), of the Production Environment (PP), of the DSMS Journal (BJ), of the DSMS Database (BB), and of the Pactable Database (TC).

This utility includes two procedures: ESES and CSES.

5.1.2. ESES: EXTRACTION OF SESSION NUMBERS

ESES: INTRODUCTION

The Extraction of Session Numbers procedure (ESES) creates a correspondence-table file linking older frozen sessions and new frozen sessions.

PRELIMINARY OPERATIONS

Backup of the VA Pac files:

```
.Archival of the Journal (ARCH)
.Backup of the VA Pac Database (SAVE)
.Backup of the Generation-Print requests file (SVAG)


If PEI is installed:.PEI backup (SVPE)


If Pactables is installed:.Table backup (SVTA)
```

If DSMS is installed, perform a backup of the DSMS environment, by:

```
.Archiving the DSMS Journal (DARC)
.Backing up the DSMS Database (DSAV)
```

EXECUTION CONDITIONS

None.

Batch procedure access authorization option: level 4 required.

USER INPUT

Batch procedure access authorization option: a '*' line with User code and Password is required.

One line per session number to force :

```
-------------------------------------------------------------
!Pos.! Lon.! Valeur ! Meaning                                !
!-------------------------------------------------------------!
!  2 !   1 ! 'S'    ! Line Code                               !
!  3 !   4 ! nnnn   ! Original session number                !
!  7 !   4 ! nnnn   ! New session number                     !
-------------------------------------------------------------
```

5.1.3. ESES: DESCRIPTION OF STEPS

ESES: DESCRIPTION OF STEPS

CREATION OF THE SESSION-NUMBER CORRESPONDENCE FILE: PTUESS

```
.Permanent input file:
 -Error-message file
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN

.Input file:
 -Input transactions
  PAC7MB   (MBCSES file in directory INPUT)

.Output file:
 -Session-number correspondence table
  PAC7MV

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

.Return code:
 8:  Unauthorized user.
```

5.1.4. ESES: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) ESES BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------"
echo "                        ESES PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : CORRESPONDING SESSION NUMBER TABLE
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PACSAVPC.ini
PAC7PC=$PACSAVPC
export PAC7PC
PAC7MB=$PACINPUT'MBESES'
export PAC7MB
PAC7MV=$PACINPUT'MVESES'
export PAC7MV
PAC7DD=$PACTMP'ESESDD.ESS'
export PAC7DD
PAC7EU=$PACTMP'ESESEU.ESS'
export PAC7EU
echo "Execution: PTUESS"
cobrun PTUESS
RETURN=$?
case $RETURN in
0)
 ;;
8)
 echo "Error in executing PTUESS"
 echo "Error 8: Unauthorized user"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PTUESS"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
exit $RETURN
```

5.1.5. CSES: COMPRESSION OF SESSION NUMBERS

CSES: INTRODUCTION

The Compression of Session Numbers procedure (CSES) compresses the session numbers of the VisualAge Pacbase Database logical backups, the Pactables Database if this module is installed on the site, and the DSMS Database if this module is installed on the site. It uses the correspondence table created by the ESES procedure. The resulting files must be restored.

EXECUTION CONDITIONS

None.

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

5.1.6. CSES: USER INPUT

CSES: USER INPUT

Batch procedure access authorization: A * line with User Code and Password.

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

The line is built as follows:

Col.	Len.	Value	Meaning
2	1	'S'	Line code
3	21		Code of the files to retrieve (PC PJ PG PP BB BJ TC) separated with a blank
33	4		If the DSMS database has to be retrieved: VA Pac database logical code

5.1.7. CSES: DESCRIPTION OF STEPS

CSES: DESCRIPTION OF STEPS

'COMPRESSION' OF SESSION NUMBERS: PTUCSS

.Permanent input files:
 -Error-message file
  PAC7AE

.Input file (from ESES procedure):
 -Session-number correspondence table
  PAC7MV

.Transaction file:
 -User input
  PAC7MB (MBCSESfile in INPUT directory)

.Retrieval of the VisualAge Pacbase database backup
 -Input
  PAC7PC
  If Dispatch option of the backup:
  PAC7PD
 -Output
  PAC7CP
  If Dispatch option of the backup:
  PAC7DP

.Retrieval of the VisualAge Pacbase archived journal:
 -Input
  PAC7PJ
 -Output
  PAC7JP

.Retrieval of the VA Pac generation-print request backup:
 -Input
  PAC7PG
 -Output
  PAC7GP

.Retrieval of the PEI backup:
 -Input
  PAC7PP

 -Output
  PAC7EP

If DSMS is installed:

.Retrieval of the DSMS database backup:
 -Input
  PACDBB
 -Output
  PACDJB

.Retrieval of the DSMS archived journal:
 -Input
  PACDDJ
 -Output
  PAC7JD

If Pactables is installed:

.Retrieval of the Pactables database backup:
 -Input
  PACDTC
 -Output
  PACDCT

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

5.1.8. CSES: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) CSES BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                        CSES PROCEDURE"
echo "                        =============="
echo "Directory 'assign'               : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : `dirname $PACINPUT.`"
echo "------------------------------------------------------------"
echo "WARNING:"
echo "If the DSMS backup files need to be retrieved,"
echo "please assign a value to the following parameters:"
echo "------------------------------------------------------------"
echo "DSMS install directory           : $2"
echo "Name of the DSMS Database        : $3"
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : COMPRESSION OF SESSION NUMBERS
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# * .COMPRESSION OF SESSION NUMBERS COMMAND LINE
# * COL  2    : 'S'
# * COL  3-19 : CODE OF THE FILES TO RETRIEVE (PC PJ
# *           : (PC PJ PG PP BB BJ) SEPARATED WITH A BLANK
# * COL 33-36 : IF THE DSMS DATABASE HES TO BE RETRIEVED :
# *           : VA Pac DATABASE LOGICAL CODE
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBCSES'
export PAC7MB
PAC7MV=$PACINPUT'MVESES'
export PAC7MV
. $PACDIR/assign/$1/PACSAVPC.ini
PAC7PC=$PACSAVPC
export PAC7PC
PAC7CP=$PACSAVPCNEW
export PAC7CP
. $PACDIR/assign/$1/PACSAVPJ.ini
PAC7PJ=$PACSAVPJ
export PAC7PJ
PAC7JP=$PACSAVPJNEW
export PAC7JP
. $PACDIR/assign/$1/PACSAVPG.ini
PAC7PG=$PACSAVPG
export PAC7PG
PAC7GP=$PACSAVPGNEW
export PAC7GP
. $PACDIR/assign/$1/PACSAVPP.ini
PAC7PP=$PACSAVPP
export PAC7PP
PAC7EP=$PACSAVPPNEW
export PAC7EP
if [ -d "$2/assign/$3/PACSAVBB.ini" ]
```

```
    then
       . $2/assign/$3/PACDBB.ini
       PAC7BB=$PACSAVBB
       export PAC7BB
       PAC7JB=$PACSAVBBNEW
       export PAC7BB
    fi
    if [ -d "$2/assign/$3/PACSAVBJ.ini" ]
    then
       . $2/assign/$3/PACDBJ.ini
       PAC7DJ=$PACSAVBJ
       export PAC7DJ
       PAC7JD=$PACSAVBJNEW
       export PAC7JD
    fi
    PAC7DD=$PACTMP'CSESDD.CSS'
    export PAC7DD
    PAC7EU=$PACTMP'CSESEU.CSS'
    export PAC7EU
    echo "Execution : PTUCSS"
    cobrun PTUCSS
    RETURN=$?
    case $RETURN in
    0)
     ;;
    8)
     echo "Error in executing PTUCSS"
     echo "Error 8 : Utilisateur non autorise"
     sh $PACDIR/batch/proc/ERRPAUSE.ini
     exit $RETURN
     ;;
    *)
     echo "Error in executing PTUCSS"
     echo "Error $RETURN"
     sh $PACDIR/batch/proc/ERRPAUSE.ini
     exit $RETURN
     ;;
    esac
    # ************************************************************
    echo "End of procedure"
    echo ""
    if [ -f "$PACSAVPCNEW" ]
    then
       echo "Call of PCBACKUP.ini file"
       sh $PACDIR/assign/$1/PCBACKUP.ini
    fi
    if [ -f "$PACSAVPJNEW" ]
    then
       echo "Call of PJBACKUP.ini file"
       sh $PACDIR/assign/$1/PJBACKUP.ini
    fi
    if [ -f "$PACSAVPGNEW" ]
    then
       echo "Call of PGBACKUP.ini file"
       sh $PACDIR/assign/$1/PGBACKUP.ini
    fi
    if [ -f "$PACSAVPPNEW" ]
    then
       echo "Call of PPBACKUP.ini file"
       sh $PACDIR/assign/$1/PPBACKUP.ini
    fi
    if [ -f "$PACSAVBBNEW" ]
    then
       echo "Call of BBBACKUP.ini file"
       sh $2/assign/$3/BBBACKUP.ini
    fi
    if [ -f "$PACSAVBJNEW" ]
    then
       echo "Call of BJBACKUP.ini file"
       sh $2/assign/$3/BJBACKUP.ini
    fi
    exit $RETURN
```

5.2. GBIR: PARTITIONED DATABASE MANAGER

5.2.1. GBIR: INTRODUCTION

GBIR: INTRODUCTION

The PARTITIONED DATABASE MANAGER (LCU-) is a utility option of the Dictionary function, and its use depends on the corresponding purchase agreement.

Users likely to use this utility are those who work with databases shared by one or more sites, and who might therefore be working on several versions of the same sub-network.

With this utility, you can align all versions of a particular sub-network, taking into account the update transactions performed on any one of these versions.

In more general terms, through the Sub-Network Comparison Utility, any two versions of a sub-network may be aligned. For example, this utility can be used when the current version of a sub-network has to take into account update transactions performed on a frozen session of this sub-network.

For additional information, refer to the OPTIONAL UTILITIES Reference Manual.

PRINCIPLES

Two methods may be used to align a 'slave' sub-network with a 'master' sub-network:

The standard method generates batch transactions which are used to update the 'slave' sub-network. The standard validations performed by the update ensure the consistency of updated data in the 'slave' sub-network.

The second method involves merging the 'master' sub-network with the network containing the 'slave' sub-network: the 'master' sub-network replaces the 'slave' sub-network. The results of the merge must be reorganized via the REOR procedure to obtain a back-up of the new network, which can be used as input to the REST procedure.

No validation is performed on data consistency. Thus, this method must only be used when standard network management ensures data consistency between the networks.

For more details, refer to the OPTIONAL UTILITIES Reference Manual.

1. ALIGNMENT THROUGH THE BATCH UPDATE PROCEDURE

The Sub-Network Comparison Utility generates an update transaction flow making a 'slave' sub-network identical to a 'master' sub-network.

This is done in two steps:

- The extraction, in sequential form, of the sub-network image, which must be aligned via the PACX procedure (EXLI extractor, formatting for CPSN). (For further details, see Chapter STANDARD PROCEDURES, Subchapter 'PACX: Extraction from the VA Pac Database', in the 'Batch Procedures, User's Guide'.)

- The comparison of images, two-by-two, in order to produce an update transaction flow (CPSN procedure).

These two operations may be executed at different sites.

NOTES ON THE GENERATED UPDATE TRANSACTION FLOW

It is logically impossible to align P.I.A.'s: for the modification of a P.I.A. in a 'master' sub-network, the generated update transactions will not be accepted if the P.I.A. is already called in a library of the 'slave' sub-network.

In the update report of the 'slave' sub-network (UPDT procedure), some '0' or 'H' lines may be rejected with the following error message:

"INVALID ABSENCE FOR THE FIELD PROGRAM NAME"

This message can be ignored; the update is executed correctly.

2. ALIGNMENT THROUGH THE SUB-NETWORK MERGE

The Sub-Network Merge Utility generates a sequential file which is the result of the merge of a 'master' sub-network into a target network. This 'master' sub-network completely replaces the 'slave' sub-network.

The replacement of the 'slave' sub-network is done on a library-to-library basis. If the library hierarchy of the 'master' sub-network is different from that of the 'slave' (new, deleted or modified libraries), the modifications must be applied to the target network via the MLIB procedure before the merge procedure.

The library codes may be different in the 'slave' and 'master' sub-networks.

The sub-network merge is executed in three steps:

. Extraction of the 'master' sub-network, whose output is a sequential file (EMSN procedure),

. Merge of the extracted sub-network with the target network (MESN procedure), yielding a merged file to be used as input to the REOR procedure,

. Reorganization of the merge result (REOR procedure), yielding a new network back-up.

These three operations may be executed at different sites.

IMPORTANT NOTE

NO consistency check on the data in the network hierarchy is performed (see paragraph "PRINCIPLES" above).

5.2.2. CPSN: SUB-NETWORK COMPARISON
5.2.2.1. CPSN: INTRODUCTION

CPSN: INTRODUCTION

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

The 'master' sub-network is used as the reference when updating the 'slave' sub-network.

EXECUTION CONDITION

Batch procedure access authorization option: Level 3 is required.

ABENDS

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

5.2.2.2. CPSN: NOTES ON THE RESULTS

USER INPUT

Batch procedure access authorization option:
One '*'line :

COL.	LEN.	VALUE	MEANING
2	1	*	LINE CODE
3	8	uuuuuuuu	USER CODE
11	8	pppppppp	USER PASSWORD
40	3	ppp	DSMS Product Code
43	6	nnnnnn	DSMS Change number
			(DSMS module only)
49	1		Lock management
		' '	Extract. of locks without user code
		'1'	No extraction of locks
		'2'	Extract. of locks with user code
50	1	' '	No transfer of the password on the * line at the top of generated trans.
		'1'	Transfer of the password on the * line at the top of generated trans.

NOTES ON THE RESULTS

The two sub-networks being compared must have been extracted via the PACX procedure (EXLI extractor, formatting for CPSN).

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

The comparison is made between libraries located in the same place in the two sub-networks, but it is not necessary for the two corresponding libraries to have the same code.

If the 'master' sub-network contains libraries that do not exist in the 'slave' sub-network, you have to initialize these libraries in the 'slave' sub-network before doing the extraction. To do this, use the MLIB procedure followed by the REST procedure.

5.2.2.3. CPSN: DESCRIPTION OF STEPS

CPSN: DESCRIPTION OF STEPS

COMPARISON OF SUB-NETWORKS: PTU850

This step compares two sub-networks with the
same hierarchical structure, one being considered as the
'master', the other as the 'slave'.

.Permanent input file:
 -Error message file
  PAC7AE

.Transaction file:
 -User input
  PAC7MB (MBCPSN file in INPUT directory)

.Input files from PACX:
 -Master sub-network
  PAC7MA (MAIN.FI file in temporary directory)
 -Slave sub-network
  PAC7ES (SLAVE.FI file in temporary directory)

.Output file:
 -Update transactions and sort criterion
  PAC7MK

.Output reports:
 -Report
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

.Return codes:
 . 0: OK.
 . 8: Error, or unauthorized user

FORMATTING GENERATED TRANSACTIONS: PTU855

This step formats the generated and sorted transactions
and prints them. It is executed when no error is found.

.Permanent input file:
 -Error message file
  PAC7AE

.Input work file:
 -Sorted generated transactions
  PAC7MK

.Output file:
 -Transactions generated for update
  PAC7MB (MBCPSN file in INPUT sub-directory)

.Output report:
 -Generated transactions
  PAC7EU

5.2.2.4. CPSN: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) CPSN BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         CPSN PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : SUB-NETWORK UTILITIES
# *                 COMPARISON OF SUB-NETWORKS
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  ONE '*' LINE WITH USER CODE AND PASSWORD
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBCPSN'
export PAC7MB
PAC7ES=$PACTMP'SLAV.FI'
export PAC7ES
PAC7MA=$PACTMP'MAIN.FI'
export PAC7MA
PAC7MK=$PACTMP'MK'
export PAC7MK
PAC7EU=$PACTMP'CPSNEU.850'
export PAC7EU
PAC7DD=$PACTMP'CPSNDD.850'
export PAC7DD
echo "Execution : PTU850"
cobrun PTU850
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 PAC7MB=$PACINPUT'MVCPSN'
 export PAC7MB
 PAC7MK=$PACTMP'MK'
 export PAC7MK
 PAC7EU=$PACTMP'CPSNEU.855'
 export PAC7EU
 echo "Execution : PTU855"
 cobrun PTU855
 RETURN=$?
 case $RETURN in
 0)
  echo "End of procedure"
  echo ""
  echo "Deletion of the temporary file"
  rm -f $PACTMP'MK'
 ;;
 *)
```

```
     echo "Error in executing PTU855"
    ;;
    esac
    ;;
   8)
    echo "Error in executing PTU850"
    echo "OR : Error on * input line"
   ;;
   *)
    echo "Error in executing PTU850"
   ;;
   esac
   if [ "$RETURN" != '0' ]
   then
      sh $PACDIR/batch/proc/ERRPAUSE.ini
   fi
   exit $RETURN
```

5.2.3. SASN: SUB-NETWORK BACKUP
5.2.3.1. SASN: INTRODUCTION

SASN: INTRODUCTION

The Sub-Network Backup procedure (SASN) extracts one or several sub-networks from a database. The result is a consistent set of libraries which will make up a new database (formatted as a backup file to be used as input to the Restoration procedure).

Each extracted sub-network is identified by its lowest-level library; the utility automatically extracts all higher-level libraries pertaining to the sub-network.

The SASN procedure may be equated with the MLIB procedure, the only difference is that the SASN procedure deletes gaps.

EXECUTION CONDITION

The database must be closed to on-line use.

Batch procedure access authorization option: Level 4 is required.

ABNORMAL EXECUTION

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

5.2.3.2. SASN: USER INPUT

SASN: USER INPUT

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

POS.	LEN.	VALUE	MEANING
1	2	' '	Not used
3	3	bbb	Code of lowest-level library of the sub-network to be extracted. (All the upper-libraries of 'bbb' will be automatically extracted.)

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

5.2.3.3. SASN: DESCRIPTION OF STEPS

SASN: DESCRIPTION OF STEPS

DATABASE VALIDATION: PTU130

This program is always executed.

.Permanent input files:
 -Error message file
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN

.Transaction input file:
 -Database-selection transactions
  PAC7MB

.Output files:
 -Sequential data image:
  PAC7RP
  (Must be able to contain all data)
 -Sequential index image
  PAC7NA
  (Must be able to contain all indexes)
 -Sequential frozen data image
  PAC7RA

.Sort file(s):
  Not assigned

.Output reports:
 -Execution report
  PAC7DS
 -Batch-procedure authorization option
  PAC7DD

.Return codes:

  - 0: OK
  - 5: At least one of the selected libraries does not
       exist
  - 6: More than 99 libraries are selected
  - 8: Unauthorized user

FORMATTING OF SEQUENTIAL IMAGE: PTU140

This program is executed when no error is found in the input transactions.

.Permanent input files:
 -Error message file
  PAC7AE

.Input work files:
 -Data sequential image
  PAC7RP
 -Index sequential image
  PAC7NA
 -Frozen data sequential image
  PAC7RA

.Output file:
 -Database sequential image
  PAC7SR
 If Dispatch option:
 -Database sequential image #2
  PAC7PD

.Sort file(s):
  Not assigned

.Output report:
 -Execution report
  PAC7DS

5.2.3.4. SASN: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SASN BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                         SASN PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : SUB-NETWORK UTILITIES
# *                 SUB-NETWORK BACKUP
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  ONE '*' LINE WITH USER CODE AND PASSWORD
# * .EXTRACTION REQUEST
# *  ONE LINE PER LIBRARY TO BE EXTRACTED
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBSASN'
export PAC7MB
PAC7NA=$PACTMP'NA'
export PAC7NA
PAC7RA=$PACTMP'RA'
export PAC7RA
PAC7RP=$PACTMP'RP'
export PAC7RP
PAC7DS=$PACTMP'SASNDS.130'
export PAC7DS
PAC7DD=$PACTMP'SASNDD.130'
export PAC7DD
echo "Execution : PTU130"
cobrun PTU130
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PACSASNPC.ini
 PAC7SR=$PACSASNPC
 export PAC7SR
 PAC7PD=$PACSASNPCI
 export PAC7PD
 PAC7NA=$PACTMP'NA'
 export PAC7NA
 PAC7RA=$PACTMP'RA'
 export PAC7RA
 PAC7RP=$PACTMP'RP'
 export PAC7RP
 PAC7DS=$PACTMP'SASNDS.140'
 export PAC7DS
```

```
 echo "Execution : PTU140"
 cobrun PTU140
 RETURN=$?
 case $RETURN in
 0)
  echo "End of procedure"
  echo ""
  echo "Deletion of the temporary files"
  rm -f $PACTMP'NA'
  rm -f $PACTMP'RA'
  rm -f $PACTMP'RP'
  ;;
 *)
  echo "Error in executing PTU140"
  ;;
 esac
 ;;
8)
 echo "Error in executing PTU130"
 echo "Error 8 : Error on * input line"
 ;;
6)
 echo "Error in executing PTU130"
 echo "Error 6 : More than 99 input transactions"
 ;;
5)
 echo "Error in executing PTU130"
 echo "Error 5 : One of the selected libr. does not exist"
 ;;
*)
 echo "Error in executing PTU130"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

5.2.4. EMSN: EXTRACTION FOR SUB-NETWORK MERGE
5.2.4.1. EMSN: INTRODUCTION

EMSN: INTRODUCTION

The Extraction for Sub-Network Merge procedure (EMSN) extracts a sub-network from a database, producing a sequential file to be used as input to the Sub-Network Merge (MESN) procedure.

EXECUTION CONDITION

None, because the database is not updated directly.

Batch procedure access authorization option: Level 3 is required.

ABENDS

In case of an abend, the procedure may be restarted as it is once the problem has been corrected.

5.2.4.2. EMSN: USER INPUT

EMSN: USER INPUT

```
One '*' line per library to extract:

-------------------------------------------------------------
! POS.! LEN.! VALUE  ! MEANING                               !
!-----------------------------------------------------------!
!  2  !  1  ! '*'    ! Line code                              !
!  3  !  8  !uuuuuuuu! User code                             !
! 11  !  8  !pppppppp! User password                         !
! 19  !  3  ! bbb    ! Library code                           !
! 22  !  4  ! ssss   ! Session number (blank=current ses.)  !
! 26  !  1  ! T      ! Session status if Test session        !
-------------------------------------------------------------
```

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

NOTES:

The number of libraries to be extracted is limited to 99.

This set of libraries is called a 'sub-network'. The order of the extraction requests must be the same as the description of the sub-network in the Inter-library (***).

The '*' lines MUST be sorted in descending order from left to right of the sub-network; the order of the requests is not checked by the system. If even one request is invalid, all the others are also rejected.

The extracted sub-network does not need to be complete.

EXAMPLE

```
LIBRARY CODE          Corresponding extraction transactions:

AAA                          AAA is not extracted
   XXX                       (1) _*USERCODEPASSWORDXXX
      DDD                    (2) _*USERCODEPASSWORDDDD
      EEE                    (3) _*USERCODEPASSWORDEEE
         KKK                 (4) _*USERCODEPASSWORDKKK
         RRR                 (5) _*USERCODEPASSWORDRRR
   MMM                       (6) _*USERCODEPASSWORDMMM
```

PRINTED OUTPUT

The EMSN procedure prints a report stating:

- The list of applied transactions,

- The list of the sub-network libraries (including libraries which were not extracted), which corresponds to the input lines which will be required in the MESN procedure.

EXAMPLE:

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

5.2.4.3. EMSN: DESCRIPTION OF STEPS

EMSN: DESCRIPTION OF STEPS

SUB-NETWORK EXTRACTION: PTU810

This step may extract up to 99 libraries.

.Permanent input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error message file
  PAC7AE

.Transaction file:
 -User input
  PAC7ME

.Output file:
 -Extracted sub-network
  PAC7BB

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

.Sort file(s):
 Not assigned

.Return codes:
 . 0: OK.
 . 8: Error or unauthorized user

The return code is set when the EMSN procedure is immediate-
ly followed by the execution of the MESN procedure.

5.2.4.4. EMSN: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) EMSN BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         EMSN PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : SUB-NETWORK UTILITIES
# *                 EXTRACTION FOR SUB-NETWORK MERGE
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE * LINE PER SUB-NETWORK LIBRARY TO BE EXTRACTED
# *  (NUMBER OF LINES LIMITED TO 99)
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7ME=$PACINPUT'MBEMSN'
export PAC7ME
PAC7BB=$PACTMP'EMSN.FI'
export PAC7BB
PAC7EE=$PACTMP'EMSNEE.810'
export PAC7EE
PAC7EU=$PACTMP'EMSNEU.810'
export PAC7EU
PAC7DD=$PACTMP'EMSNDD.810'
export PAC7DD
echo "Execution : PTU810"
cobrun PTU810
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
8)
 echo "Error in executing PTU810"
 echo "OR : Error on * input line"
 ;;
*)
 echo "Error in executing PTU810"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

5.2.5. MESN: SUB-NETWORK MERGE
5.2.5.1. MESN: INTRODUCTION

MESN: INTRODUCTION

Through the MESN procedure, one sub-network may be replaced by another sub-network extracted via the EMSN procedure.

The extracted sub-network deletes and replaces the corresponding sub-network in the Database back-up, providing a merged file which, when reorganized via REOR, will become the back-up of the new database.

THERE IS NO CONSISTENCY CHECK ON THE NEW DATABASE. THIS PROCEDURE MUST BE USED ONLY IN CASES WHERE CURRENT MANAGEMENT OF DATABASES AND SUB-NETWORKS BY THE USER ENSURES DATA CONSISTENCY.

EXECUTION CONDITION

This procedure must be preceded by the EMSN procedure, which extracts the sub-network to be merged.

The 'master' sub-network and the 'slave' sub-network must have exactly the same library hierarchy.

Batch procedure access authorization option: Level 4 is required.

ABENDS

In case of an abend, the procedure can be restarted as it is once the problem is corrected.

PRINTED OUTPUT

The procedure prints a merge report.

When input transactions do not correspond to the libraries found in the extracted sub-network, error messages are displayed, but the procedure is correctly executed.

5.2.5.2. MESN: USER INPUT

MESN : USER INPUT

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

One '*' line is required for each library of the sub-network, including those which are not extracted.

These lines must be coded according to the output of the EMSN procedure and, when required, with the code of the corresponding 'slave' sub-network library.

All sub-network libraries, including those which have not been extracted, must be indicated.

POS.	LEN.	VALUE	MEANING
1	1	'*'	Library not extracted
		'R'	Extracted library
2	1	'*'	Line code
3	3	aaa	'Master' sub-network library code (Required)
6	3	bbb	'Slave' sub-network library code (Default option: 'master' sub-network library code)

In case of error, the procedure is interrupted.

Example of User Input

Without code modifications:

```
**AAA
R*XXX
R*DDD
R*EEE
R*KKK
R*RRR
R*MMM
```

With code modifications:

```
**AAACEN
R*XXXAPP
R*DDD
R*EEEBIB
R*KKK
R*RRR
R*MMM
```

Although the AAA library was not extracted, the corresponding input line must be entered, with the code of the corresponding library in the target network, if it is not AAA (CEN in this example).

5.2.5.3. MESN: DESCRIPTION OF STEPS

MESN: DESCRIPTION OF STEPS

SUB-NETWORK MERGE: PTU815

This step merges the sub-network extracted via the EMSN procedure with the target network.

.Permanent input files:
 -Backup file to merge
  PAC7PC
 -Extracted sub-network
  PAC7BB
 -Error message file
  PAC7AE

.Transaction file:
 -User input
  PAC7ME

.Output file:
 -Merge file to be reorganized
  PAC7CP

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

.Return code:
  -8: Unauthorized user

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

5.2.5.4. MESN: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) MESN BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-------------------------------------------------------"
echo "                        MESN PROCEDURE"
echo "                        =============="
echo "Directory 'assign'               : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                  : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : SUB-NETWORK UTILITIES
# *                 SUB-NETWORK MERGE
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  ONE '*' LINE WITH USER CODE AND PASSWORD
# * .MERGE REQUEST
# *  ONE '*' LINE FOR EACH LIBRARY OF THE SUB-NETWORK TO
# *  BE MERGED (MAXIMUM: 99 LINES).
# *  COL. 1    : 'R' LIBRARY TO BE MERGED
# *            : '*' HIGHER LEVEL LIBRARY NOT MERGED
# *  COL. 2    : '*'
# *  COL. 3-5  : LIBRARY CODE IN THE 'MASTER' NETWORK
# *  COL. 6-8  : TARGET LIBRARY CODE IN THE 'SLAVE' NETWORK
# *            : (DEFAULT OPTION: 'MASTER' NETWORK LIBRARY CODE)
# *
# *  A LIST OF THE INPUT LINES FOR THIS PROCEDURE IS PRINTED
# *  AS AN OUTPUT OF THE EMSN PROCEDURE.
# *  IF A LIBRARY HAS A DIFFERENT NAME IN THE NEW, MERGED
# *  NETWORK, THIS NAME MUST BE ENTERED IN THE CODE OF
# *  THE TARGET LIBRARY.
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PAC7ME=$PACINPUT'MBMESN'
export PAC7ME
. $PACDIR/assign/$1/PACSAVPC.ini
PAC7PC=$PACSAVPC
export PAC7PC
. $PACDIR/assign/$1/PACMESNPC.ini
PAC7CP=$PACMESNPC
export PAC7CP
PAC7BB=$PACTMP'EMSN.FI'
export PAC7BB
PAC7EU=$PACTMP'MESNEU.815'
export PAC7EU
PAC7DD=$PACTMP'MESNDD.815'
export PAC7DD
echo "Execution : PTU815"
cobrun PTU815
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
```

```
 ;;
8)
 echo "Error in executing PTU815"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PTU815"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

5.3. LOAE: AE - AP RELOADING

5.3.1. LOAE: INTRODUCTION

LOAE: INTRODUCTION

The LOAE procedure restores the AE and AP indexed files when one of them (or both) is physically lost.

Restoration is performed from the last backup of the user parameters (PE file), and from the error message file (AE0).

EXECUTION CONDITION

On-line access to the AE and AP file must be closed.

ABNORMAL EXECUTIONS

Refer to Chapter "OVERVIEW", Subchapter 'ABNORMAL ENDINGS', for more details.

5.3.2. LOAE: USER INPUT

LOAE: USER INPUT

One compulsory line:

Pos.	Len.	Value	Meaning
2	6	'NRREST'	Line code

5.3.3. LOAE: DESCRIPTION OF STEPS

LOAE: DESCRIPTION OF STEPS

LOADING OF THE AE AND AP FILES: PACU80

.Permanent input files:
 -User parameter backup
  PAC7CE
 -Initial sequential image of error messages
  PAC7LE

.Transaction file:
 -Update transactions
  PAC7MC (MBLOAE file in INPUT directory)

.Permanent output files:
 -Error messages
  PAC7AE
 -User parameters
  PAC7AP

.Sort file(s):
  Not assigned

.Output report:
 -Reconstruction report
  PAC7IJ

5.3.4. LOAE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) LOAE BATCH Procedure
#@(#)
# Controle des parametres
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        LOAE PROCEDURE"
echo "                        =============="
echo "Repertoire 'assign'               : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Repertoire 'tmp'                  : »dirname $PACTMP»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Repertoire 'input'                : »dirname $PACINPUT»"
if [ -n "$2" ]
then
   echo "Radical fichiers 'tmp' et 'input' : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : AE AND AP FILE LOADING
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AP.ini
PAC7MC=$PACINPUT'MBLOAE'
export PAC7MC
. $PACDIR/assign/$1/PACSAVAE0.ini
PAC7LE=$PACSAVAE0
export PAC7LE
. $PACDIR/assign/$1/PACSAVPE.ini
PAC7CE=$PACSAVPE
export PAC7CE
PAC7IJ=$PACTMP'LOAEIJ.U80'
export PAC7IJ
echo "Execution: PACU80"
cobrun PACU80
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PACU80"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
exit $RETURN
```

5.4. VINS: INSTALLATION OF THE VA SMALLTALK DICTIONARY

5.4.1. VINS: INTRODUCTION

VINS: INTRODUCTION

VINS: INSTALLATION

The VINS procedure performs a batch update of the database, based on transactions provided with the product. It is used for the installation of the VA Pac/VA Smalltalk and VA Pac/TeamConnection bridges.

Entities are created in Inter-Library mode, which allows access from any Library of the network.

If some user entities have the same codes in the sub-network, VINS refuses to create them in inter-library mode, except if the update option has been set to 'F' on the '*' line. In such a case, VINS deletes all user entities with this code in the sub-network. A report then lists the user entities that have been deleted. The corresponding deletion transactions are not journalized.

EXECUTION CONDITION

On-line access must be prohibited.

Global authorization level 4 is required.

ABENDS

Refer to chapter 'OVERVIEW', sub-chapter 'Abnormal Endings'.

When the abend occurs during the execution of the PACINS program, the database is no longer consistent. Once the problem is solved, the database must be re-loaded with the retrieval of the archived transactions. The VINS procedure must then be executed again.

5.4.2. VINS: USER INPUT

VINS: INPUT-PROCESSING-RESULTS

USER INPUT

The VINS procedure requires two types of user input.

. User ID:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8		User code
11	8		Password
27	1		Update option:
			' ' - No update
			'S' - Update simulation with printing of list of U.E.'s to be cancelled
			'F' - Forcing the cancellation of U.E.'s with the same codes in lower level libraries

. Transactions used to create the necessary User Entities, which are provided on installation: the contents of these transactions MUST NOT BE MODIFIED.

PRINTED OUTPUT

The procedure prints out:

- A global report of the update,

- If the update option was set, the list of cancellation transactions.

RESULT

Once the update is performed, the network is ready for either on line or batch use.

5.4.3. VINS: DESCRIPTION OF STEPS

VINS: DESCRIPTION OF STEPS

DATABASE UPDATE: PACINS

.Permanent update files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Journal file
  PAC7AJ

.Permanent input file:
 -Error message file
  PAC7AE

.Input-transaction files:
 -User-Entity transactions
  PAC7MV
 -'*' line transaction
  PAC7MB

.Output reports:
 -Update report
  PAC7IE
 -Deletion-transaction list
  PAC7EE
 -Batch-procedure error report
  PAC7DD

5.4.4. VINS: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) VINS BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        VINS PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
. $PACDIR/assign/$1/PAC7AJ.ini
echo "Directory of the AJ file          : »dirname $PAC7AJ.»"
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : VISUAL ENTITIES DICTIONARY UPDATING
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AJ.ini
PAC7MB=$PACINPUT'MBVINS'
export PAC7MB
PAC7MV=$PACINPUT'MVVINS'
export PAC7MV
PAC7DD=$PACTMP'VINSDD.INS'
export PAC7DD
PAC7EE=$PACTMP'VINSEE.INS'
export PAC7EE
PAC7IE=$PACTMP'VINSIE.INS'
export PAC7IE
echo "Execution: PACINS"
cobrun PACINS
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PACINS"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
exit $RETURN
```

5.5. RTLO: DELETION OF INVALID UPDATE LOCKS

5.5.1. RTLO: INTRODUCTION

RTLO: INTRODUCTION

The RTLO procedure deletes erroneous update locks produced by the retrieval of a previous release of the Database.

The problem is detected by the fact that an ENTITY TO BE CREATED is considered as an ENTITY LOCKED UNDER ANOTHER USER CODE. Such may be the case with Databases in which entities locked in frozen sessions have been deleted.

CHARACTERISTICS

This procedure does not entail any user input. It provides a stream of batch deletion transactions for invalid locks in the database, which is to be used as input to the Database Updating (UPDT) procedure.

EXECUTION CONDITION

On-line access must be closed.

PRINTED OUTPUT

This procedure prints out a list of the deleted invalid locks and a list of the generated batch deletion transactions.

5.5.2. RTLO: DESCRIPTION OF STEPS

RTLO: DESCRIPTION OF STEPS

RETRIEVAL OF INVALID LOCKS: PTULOI

.Permanent Input files:
 -Error-message file
  PAC7AE

.Permanent Input/Output files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN

.Output file:
 -Generated deletion transactions
  PAC7MB

.Output report:
 -Lists
  PAC7EU

.Internal Sort:
  Not assigned

5.5.3. RTLO: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) RTLO BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        RTLO PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : RETRIEVAL OF LOCKED ENTITIES
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MVRTLO'
export PAC7MB
PAC7EU=$PACTMP'RTLOEU.LOI'
export PAC7EU
echo "Execution : PTULOI"
cobrun PTULOI
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
*)
 echo "Error in executing PTULOI"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

5.6. UXSR: PARTIAL SUB-NETWORK EXTRACTION

5.6.1. UXSR: INTRODUCTION

UXSR: INTRODUCTION

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

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

It is also possible to select:

. A frozen session (nT):

This frozen session will become the current session in the new Database.
No other frozen session will be selected.
The image of this Database will be identical to the view which existed in the nT frozen session, but this time it will be in n+1 current session.

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

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

EXAMPLES:

. Creation of Libraries:

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

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

. Merging of Libraries in the same Library:

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

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

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

. Renaming of Library:

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

(1) Creation of the CEN Library with the contents of APP.

WARNING

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

EXECUTION CONDITION

On-line access must be prohibited. This procedure processes data only. It must therefore be followed by the REOR, then REST procedures, in order for the new Database to be taken into account.

5.6.2. UXSR: USER INPUT

UXSR: USER INPUT

One '*' line:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	4	nnnn	Session number (blank=current)
26	1	'T'	If selection of frozen session
		' '	If selection of current session
49	1		Option of locks extraction:
		' '	Locks extraction: user code = user code of '*' line
		'1'	No extraction of locks
		'2'	Locks extraction: user code = source user code
67	1	'T'	If col 26 = ' ' then selection of all the frozen session
		' '	If col 26 = ' ' then selection of the current session only

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

Pos.	Len.	Value	Meaning
1	1	'C'	Creation
2	1	'*'	Line code
3	3	bbb	Code of Library to be created
6	3	ccc	Code of higher Library if any
9	3	ddd	Code of source Library required even when creating a new Library, in this case enter any code not existing in the source Database.

NOTE: Do not use the character '*' in Library codes (incompatibility with the WorkStation).

5.6.3. UXSR: DESCRIPTION OF STEPS

UXSR: DESCRIPTION OF STEPS

FORMATTING OF THE SEQUENTIAL IMAGE: UTIXSR

.Permanent input files:
 -Data file
  PAC7AR
 -Error-message file
  PAC7AE

.Input transaction file:
 -Update transactions
  PAC7MB

.Output file:
 -Sequential image of the database
  PAC7PC

.Output reports:
 -List of user transactions
  PAC7EV
 -Resulting Database-condition
  PAC7EU
 -Batch-procedure authorization option
  PAC7DD

5.6.4. UXSR: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) UXSR BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                        UXSR PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : SUBNETWORK PARTIAL EXTRACTION
# ************************************************************
# * .* LINE (REQUIRED)
# * COL  2     : '*' LINE CODE
# * COL  3-10  : uuuuuuuu USER CODE
# * COL 11-18  : pppppppp PASSWORD
# * COL 22-25  : ssss     SESSION NUMBER
# *             :         (BLANK=CURRENT)
# * COL 26     : ' '      SESSION STATUS
# *             : 'T'
# * COL 49     : ' '      LOCKS EXTRACTION : USER CODE  =
# *             :         USER CODE CODE OF * LINE
# *             : '1'      NO EXTRACTION OF LOCKS
# *             : '2'      LOCKS EXTRACTION: USER CODE =
# *             :         SOURCE USER CODE
# * COL 67     : 'T'      IF COL 26 = ' ' THEN SELECTION OF ALL
# *             :         FROZEN SESSIONS
# *             : ' '      IF COL 26 = ' ' THEN SELECTION OF THE
# *             :         CURRENT SESSION ONLY
# ************************************************************
# * .AS MANY LINES (OPTIONAL) AS LIBRARIES TO BE EXTRACTED FOR
# *  UPDATE
# * COL  1     : 'C' CREATION
# * COL  2     : '*' LINE CODE
# * COL  3-5   : bbb CODE OF LIBRARY TO BE CREATED
# * COL  6-8   : ccc CODE OF LIBRARY IF ANY
# * COL  9-11  : ddd CODE OF THE SOURCE LIBRARY
# ************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBUXSR'
export PAC7MB
. $PACDIR/assign/$1/PACUXSRPC.ini
PAC7PC=$PACUXSRPC
export PAC7PC
PAC7EU=$PACTMP'UXSREU.XSR'
export PAC7EU
PAC7EV=$PACTMP'UXSREV.XSR'
export PAC7EV
PAC7DD=$PACTMP'UXSRDD.XSR'
export PAC7DD
```

```
echo "Execution : UTIXSR"
cobrun UTIXSR
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing UTIXSR"
 echo "ERREUR 12 : System Error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing UTIXSR"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
exit $RETURN
```

6. MIGRATIONS

6.1. CRYP: ENCRYPTION / DECRYPTION OF PASSWORDS

6.1.1. CRYP: INTRODUCTION

CRYP: INTRODUCTION

The CRYP procedure performs the encryption and decryption of user passwords in the PE user-parameter backup file.

The objective of this procedure is to transfer the PE file onto platforms with different codings.

EXECUTION CONDITION

Authorization level '4' for the update of user parameters (PARM).

6.1.2. CRYP: USER INPUT

CRYP: USER INPUT

A '*' line with the user code and the password must be entered.

The user code specified on the '*' line must exist in the PE file to be processed.

The procedure's specific user input allows for the selection of either Encryption or Decryption.

Pos.	Len.	Value	Meaning
3	6	'CODE'	Password encryption
		'DECODE'	Password decryption

NOTE: When decrypting, the backup obtained must not be reloaded via the 'PARM' procedure. If it were, user passwords would no longer be recognized.

6.1.3. CRYP: DESCRIPTION OF STEPS

CRYP : DESCRIPTION OF STEPS

ENCRYPTION / DECRYPTION OF PASSWORDS: PACU99

.Input files:
 -User parameter backup
  PAC7CE
 -User input
  PAC7MB

.Output file:
 -User parameter backup
  PAC7EC

.Output report:
 -Execution report
  PAC7DD

6.1.4. CRYP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) CRYP BATCH Procedure
#@(#)
clear
echo""
echo "-------------------------------------------------------"
echo "                        CRYP PROCEDURE"
echo "                        =============="
echo "  Please note the specific parameters:"
echo ""
echo "   Input PE : complete directory and filename of"
echo "            : PE input file (to be coded or uncoded)"
echo "            : $1"
echo "  Output PE : complete directory and filename of"
echo "            : PE output file"
echo "            : $2"
echo "Transaction : sequential file directory"
echo "            : $3"
echo "     Report : temporary file directory"
echo "            : $4"
echo ""
echo "  Example"
echo "  PROCCRYP  $PACDIR/save/PE.MVS  $PACDIR/save/PE"
echo "            $PACDIR/input/B0  $PACDIR/tmp/B0"
echo "-------------------------------------------------------"
echo""
if [ "$#" != 4 ]
then
   echo "Incorrect number of parameters"
   exit 20
fi
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : CRYPTAGE - DECRYPTAGE OF USER PASSWORDS
# ************************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# * .ENCRYPTION OR DECRYPTION OF USER PASSWORDS
# *  COL  3-8  : 'DECODE' FOR DECRYPTION
# *            : 'CODE'   FOR ENCRYPTION
# ************************************************************
PAC7CE=$1
export PAC7CE
PAC7EC=$2
export PAC7EC
PAC7MB=$3/MBCRYP
export PAC7MB
PAC7DD=$4/CRYPDD.U99
export PAC7DD
echo "Execution : PACU99"
cobrun PACU99
RETURN=$?
case $RETURN in
0)
 echo "-------------------------------------------------------"
 echo "End of procedure"
 echo ""
 echo "Output PE ($2) will be input file of procedures:"
 echo "PROCPE80 if the file is in 8.0.1 format"
 echo "PROCPARM or PROCLOAE if the file is in the right format"
 echo "-------------------------------------------------------"
 ;;
*)
```

```
     echo "Error in executing PACU99"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

6.2. LVBL: REPLACING LOW-VALUES WITH BLANKS IN PC FILE

6.2.1. LVBL: INTRODUCTION

LVBL: INTRODUCTION

The LVBL procedure inserts a blank wherever a low-value is present in the PC Database backup file.

The purpose of this procedure is to transfer the PC file onto different platforms while avoiding problems due to the presence of low-values at the time of transfer.

UTILIZATION OPTION

The LVBL procedure allows you to keep only records of the 'data' type. See the 'Description of Steps' section for further details on the implementation of this option.

EXECUTION CONDITION

None

6.2.2. LVBL: DESCRIPTION OF STEPS

LVBL: DESCRIPTION OF STEPS

REPLACEMENT OF LOW-VALUES WITH BLANKS: PTULVB

```
.EXEC line:

 Specify PARM=DATA in order to keep only the
 'data'-type records in the output file.
 To keep both 'index' and 'data' records, do not specify
 anything.

.Input file:
 -Database backup
  PAC7MC

.Output file:
 -Database backup
  PAC7PC
```

6.2.3. LVBL: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) LVBL BATCH Procedure
#@(#)
clear
echo ""
echo "------------------------------------------------------------"
echo "                         LVBL PROCEDURE"
echo "                         =============="
echo "  Please note the specific parameters:"
echo ""
echo "  PC input : complete directory and filename of PC file"
echo "           : $1"
echo "  PC output: complete directory and filename of PC file"
echo "           : $2"
echo ""
echo "  Example"
echo "      PROCLVBL  $PACDIR/save/B0/PC.LOW $PACDIR/save/B0/PC"
echo "------------------------------------------------------------"
echo ""
if [ "$#" != 2 ]
then
   echo "Incorrect number of parameters"
   exit 20
fi
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : REPLACING LOW-VALUES BY SPACE IN A PC FILE
# *
# ************************************************************
PAC7MC=$1
export PAC7MC
PAC7PC=$2
export PAC7PC
echo "Execution : PTULVB"
cobrun PTULVB
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
*)
 echo "Error in executing PTULVB"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

6.3. SMTD: BACKUP OF TABLE DESCRIPTIONS FOR MIGRATION

6.3.1. SMTD: INTRODUCTION

SMTD: INTRODUCTION

The SMTD procedure backs up the TD table-description file by transforming binary characters into their display format.

The aim of the procedure is to transfer the TD file onto different platforms while avoiding problems caused by the presence of these characters at the time of transfers.

EXECUTION CONDITION

None.

USER INPUT

None.

6.3.2. SMTD: DESCRIPTION OF STEPS

SMTD: DESCRIPTION OF STEPS

TD BACKUP: PTASVD

.Permanent input file:
 -Table-description file
  PAC7TD

.Output file:
 -Table-description backup for migration
  PAC7TC

6.3.3. SMTD: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SMTD BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "------------------------------------------------------------"
echo "                         SMTD PROCEDURE"
echo "                         ==============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "------------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : BACKUP OF TABLES DESCRIPTIFS
# ************************************************************
. $PACDIR/assign/$1/PAC7TD.ini
. $PACDIR/assign/$1/PACSAVPD.ini
PAC7TC=$PACSAVPDNEW
export PAC7TC
echo "Execution : PTASVD"
cobrun PTASVD
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTASVD"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
echo "Call of PDBACKUP.ini file"
. $PACDIR/assign/$1/PDBACKUP.ini
exit $RETURN
```

6.4. RMTD: RESTORATION OF TABLE DESCRIPTIONS

6.4.1. RMTD: INTRODUCTION

RMTD: INTRODUCTION

The RMTD procedure restores the TD tables description file from its TC sequential backup produced by the SMTD procedure.

This procedure entails no execution condition and no user input.

6.4.2. RMTD: DESCRIPTION OF STEPS

RMTD: DESCRIPTION OF STEPS

TD FILE RESTORATION: PTARSD

```
.Input backup file:
 -Table-Description sequential file
  PAC7TC

.Output file:
 -Table-Description file
  PAC7TD
```

6.4.3. RMTD: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) RMTD BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                         RMTD PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : »dirname $PACTMP.»"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : »dirname $PACINPUT.»"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : RESTORATION OF TABLES DESCRIPTIFS
# ************************************************************
. $PACDIR/assign/$1/PAC7TD.ini
. $PACDIR/assign/$1/PACSAVPD.ini
PAC7TC=$PACSAVPD
export PAC7TC
echo "Execution : PTARSD"
cobrun PTARSD
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
*)
 echo "Error in executing PTARSD"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

6.5. RPTD: TABLE DESCRIPTIONS RETRIEVAL

6.5.1. RPTD: INTRODUCTION

RPTD: INTRODUCTION

The RPTD procedure must be used to retrieve the TD backup file from a previous release, so as to make it usable by the RMTD, Rel. 2.0, restoration procedure.

RPTD adds the century mark to all dates used in table-descriptions handling. The pivot year for century change must be parameterized.

EXECUTION CONDITION

None.

PRINTOUT

The RPTD procedure prints a report on the retrieval.

6.5.2. RPTD: USER INPUT

USER INPUT

.One parameter line defining the pivot year for adding
the century mark.

Pos.	Len.	Value	Meaning
1	2	2 digits other than '00'	Pivot Year

6.5.3. RPTD: DESCRIPTION OF STEPS

RPTD : DESCRIPTION OF STEPS

2.0 RETRIEVAL OF TD FILE: PTAR20

```
.Input files:
 -Table-descriptions backup
  PAC7TC
 -User parameter-line
  PAC7MB

.Output file:
 -2.0 backup of table-descriptions
  PAC7TR

.Output report:
 -Retrieval report
  PAC7ET
```

6.5.4. RPTD: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) RPTD BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "-----------------------------------------------------------"
echo "                         RPTD PROCEDURE"
echo "                         =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
if [ -n "$2" ]
then
   echo "Radical 'tmp' and 'input' files   : $2"
fi
echo "-----------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : RETRIEVAL OF TABLES DESCRIPTIFS 8.XX OR 1.2
# ************************************************************
. $PACDIR/assign/$1/PACSAVPD.ini
PAC7TC=$PACSAVPD
export PAC7TC
PAC7TR=$PACSAVPDNEW
export PAC7TR
PAC7MB=$PACINPUT'MBRPTD'
export PAC7MB
PAC7ET=$PACTMP'RPTDET.R20'
export PAC7ET
echo "Execution: PTAR20"
cobrun PTAR20
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTAR20"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
echo "Call of PDBACKUP.ini file"
. $PACDIR/assign/$1/PDBACKUP.ini
exit $RETURN
```

6.6. PEAS: ASCII SORT OF USER PARAMETERS

6.6.1. PEAS: INTRODUCTION

PEAS: INTRODUCTION

The PEAS sorts the user parameter backup file (PE) as an ASCII sequence. It thus makes it possible to use this backup on ASCII platforms.

This procedure does not require any execution condition nor user input.

6.6.2. PEAS: DESCRIPTION OF STEPS

PEAS: DESCRIPTION OF STEPS

ASCII SORT ON PE FILE: PTU903

.Input backup file:
 -Original user parameters
  PAC7IN

.Output backup file:
 -User parameters sorted in ASCII sequence
  PAC7OU

6.6.3. PEAS: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) PEAS BATCH Procedure
#@(#)
clear
echo ""
echo "------------------------------------------------------------"
echo "                          PEAS PROCEDURE"
echo "                          =============="
echo "  Please note the specific parameters:"
echo ""
echo "  PE input  : complete directory and filename of PE file"
echo "            : $1"
echo "  PE output : complete directory and filename of PE file"
echo "            : $2"
echo ""
echo "  Example"
echo "  PROCPEAS  $PACDIR/save/B0/PEMVS $PACDIR/save/B0/PE"
echo "------------------------------------------------------------"
echo ""
if [ "$#" != 2 ]
then
   echo "Incorrect number of parameters"
   exit 20
fi
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PE file ascii sort
# ************************************************************
PAC7IN=$1
export PAC7IN
PAC7OU=$2
export PAC7OU
echo "Execution : PTU903"
cobrun PTU903
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTU903"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
exit $RETURN
```

6.7. PGAS: ASCII SORT OF GENERATION COMMANDS

6.7.1. PGAS: INTRODUCTION

PGAS: INTRODUCITON

The PGAS procedure sorts the generation-request backup file (PG) as an ASCII sequence. It thus makes it possible to use this backup on ASCII platforms.

This procedure does not require any execution condition nor user input.

6.7.2. PGAS: DESCRIPTION OF STEPS

PGAS: DESCRIPTION OF STEPS

ASCII SORT ON PG FILE: PTU906

.Input backup file:
-Original generation requests
PAC7IN

.Output backup file:
-Generation requests sorted as an ASCII sequence
PAC7OU

6.7.3. PGAS: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) PGAS BATCH Procedure
#@(#)
clear
echo ""
echo "------------------------------------------------------------"
echo "                          PGAS PROCEDURE"
echo "                          =============="
echo "  Please note the specific parameters:"
echo ""
echo "  PG input  : complete directory and filename of PG file"
echo "            : $1"
echo "  PG output : complete directory and filename of PG file"
echo "            : $2"
echo ""
echo "  Example"
echo "  PROCPGAS  $PACDIR/save/B0/PGMVS $PACDIR/save/B0/PG"
echo "------------------------------------------------------------"
echo ""
if [ "$#" != 2 ]
then
   echo "Incorrect number of parameters"
   exit 20
fi
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PG file ascii sort
# ************************************************************
PAC7IN=$1
export PAC7IN
PAC7OU=$2
export PAC7OU
echo "Execution : PTU906"
cobrun PTU906
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTU906"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
exit $RETURN
```

6.8. PPAS: ASCII SORT OF ENVIRONMENTS

6.8.1. PPAS: INTRODUCTION

PPAS: INTRODUCTION

The PPAS procedure sorts the environment backup file (PP) as an ASCII sequence. It is then possible to use this backup on ASCII platforms.

This procedure does not require any execution condition nor user input.

6.8.2. PPAS: DESCRIPTION OF STEPS

PPAS: DESCRIPTION OF STEPS

ASCII SORT ON PP FILE: PTU907

.Input backup file:
 -Original environments
  PAC7IN

.Output backup file:
 -Environments sorted as an ASCII sequence
  PAC7OU

6.8.3. PPAS: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) PPAS BATCH Procedure
#@(#)
clear
echo ""
echo "------------------------------------------------------------"
echo "                          PPAS PROCEDURE"
echo "                          =============="
echo "  Please note the specific parameters:"
echo ""
echo "  PP input  : complete directory and filename of PP file"
echo "            : $1"
echo "  PP output : complete directory and filename of PP file"
echo "            : $2"
echo ""
echo "  Example"
echo "  PROCPPAS  $PACDIR/save/B0/PPMVS $PACDIR/save/B0/PP"
echo "------------------------------------------------------------"
echo ""
if [ "$#" != 2 ]
then
   echo "Incorrect number of parameters"
   exit 20
fi
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ************************************************************
# * VA Pac : PP file ascii sort
# ************************************************************
PAC7IN=$1
export PAC7IN
PAC7OU=$2
export PAC7OU
echo "Execution : PTU907"
cobrun PTU907
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTU907"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
exit $RETURN
```