IBM

VisualAge Pacbase 2.5

VA PAC 2.5 : UNIX
OPERATIONS MANUAL VOLUME III : USER'S GUIDE

DELIX003252A

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.ibm.com/software/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.

Second Edition (January 2000)

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.ibm.com/software/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. GENERAL INTRODUCTION TO THE BATCH PROCEDURES

1.1. INTRODUCTION TO THE BATCH PROCEDURES USER'S GUIDE

FOREWORD

This manual documents the batch procedures that all VisualAge Pacbase users are likely to use.

These procedures first include all standard procedures dedicated to updating, generating, printing, and extracting.

They also include the procedures dedicated to the following functionalities:

. Personalized extraction and automated documentation,

. Quality analysis and control (PQC),

. Integrity checks on Methodology occurrences (associated with the VA Pac WorkStation's Pacdesign module for SSADM and YSM),

. Pac/Impact,

. VisualAge Java/Smalltalk <> VisualAge Pacbase bridge.

1.2. OVERVIEW OF THE PROCEDURES

PRESENTATION OF THE PROCEDURES

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

The following elements are included for each procedure:

. A general introduction including:

  - an Introduction,
  - the Execution conditions,
  - Abends.

. the description of the User Input, processes and Results obtained, as well as possible recommendation for use.

. the Description of Steps.

A user must have authorization to a procedure on a given database.

The user, for example, must have autorization 4 to manage the Database (MLIB, REST, etc.), and autorization 2 to extract elements from it (PACX, etc.).

Each user has:

- a general level of rights to the batch procedures,
- a right level per database (for the platforms allowing management of several user databases for a same system).

For more details, refer to the 'Batch Procedures' manual: Administrator's Guide'.

1.3. 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	'T'	Test session
		'H'	Frozen session
27	1		With the UPDT procedure, in case of multiple deletion:
		'N'	Print all transactions 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 the execution 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 the VisualAge Pacbase Interface User's Guide, Chapter 'USING THE SYSTEM ON-LINE', Subchapter 'Conversation Initialization/ Sign-on'.

1.4. PLATFORM SPECIFIC ELEMENTS

MONITOR START-UP COMMANDS

These commands (pacwst, pacwstsock, paclink, pactp, pacparm, pacbatch) start-up the monitors running on the workstations or on the on-line and batch servers.

They are described in the chapter "Monitors start-up"

1.5. STRUCTURE OF PROCEDURES COMMANDS 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 ACDIR/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.
```

1.6. ADVICE ON USE

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.

1.7. PROCEDURES START-UP

BATCH PROCEDURES START-UP

The batch procedures must be submitted from a UNIX machine.

1.8. ABNORMAL EXECUTION

ABNORMAL 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 table below indicates standard values for the status and type of operation.

NN	STATUS
21	SEQUENCE ERROR
22	DUPLICATE KEY
23	NO RECORD FOUND
24	BOUNDARY VIOLATION
30	SYSTEM ERROR
34	BOUNDARY VIOLATION (SEQ.)
35	FILE NOT FOUND
92	LOGIC ERROR (FOR EX. OPEN AN ALREADY OPENED FILE)
93	LOCKED FILE
95	INVALID OR INCOMPLETE FILE DEFINITION

OO	OPERATION
W	WRITE
RW	REWRITE
RU	READ UP
OP	OPEN
CL	CLOSE
D	DELETE
R	READ
P	START
RN	READ NEXT

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 with 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 the VisualAge Pacbase Technical Support and save all listings that could help analyze the problem.

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.

1.9. 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. STANDARD PROCEDURES

2.1. UPDT: DATABASE UPDATE

2.1.1. UPDT: INTRODUCTION

UPDT: INTRODUCTION

The Database update procedure (UPDT) executes a batch update of the database. It allows access to ALL libraries which make up the database according to the different user authorizations.

With the DSMS facility (DSM), this procedure reads the VisualAge Pacbase Entity file (DC).

EXECUTION CONDITIONS

This procedure updates the database. The AR, AN and AJ files must be closed to on-line use, except for those hardware environments that support concurrent on-line and batch access.

IMPORTANT NOTES

1. For very large updates (in terms of number of transactions, about 5000), it may be necessary to

   . Back up, archive and restore the database to increase file space or to physically reorganize the files in order to make all the free space initially provided available.

   . Temporarily suppress Journalization

   (See Chapter DATABASE MANAGEMENT, Subchapter 'Database Restoration', in the Administrator's Guide.)

2. This procedure updates the current session number in two cases:

   . When it is the first connection of the day to the Database, and

   . When it contains a Database Freeze request.

ABNORMAL EXECUTION

Refer to the Administrator's Guide, Chapter 'OVERVIEW', Subchapter 'ABNORMAL ENDINGS'.

There are two types of abnormal executions:

1) Abnormal execution occurring before the execution of the PACA15 program, or during the opening of files in this program. The procedure can be restarted after the problem is corrected.

2) Abnormal execution occurring during execution of the PACA15 program. The database is left in an inconsistent state. If the problem appeared during input-output on a database file, the printed error message and the file status will dictate the solution.

In either case, a restart can only take place after a restore using the Back-up file including the transactions archived subsequent to this back-up (REST procedure).

2.1.2. UPDT: UPDATE RULES - RESULTS

UPDT: UPDATE RULES - RESULTS

USER INPUT

Refer to the bordereaux batch updates and to the input description corresponding to each entity.

The *-type line for user identification contains the user code, password and the corresponding library. It can also contain indications on the language used and the conversion.

If the update transactions correspond to an extraction, the * line generated by the extraction procedure has a language code in column 28 in order to effectively interpret the deletion action code (A in French, D in English).

A 'N' in column 67 suppresses the Lowercase-Uppercase conversion.

Pos.	Length	Value	Meaning
28	1		Language code, useful when transactions are not in the same language as the database.
		'A'	English
		'F'	French
67	1	'N'	Uppercase/Lowercase conversion deactivation.

UPDATE RULES

Each set of transactions for a library must be preceded by a *-type line.

Update transactions are not sorted.

- DATABASE FREEZE:

The 'X1HIST' specific request allows to freeze a session.

With the 'X1HIST' card, a comment can be inserted between columns 8 and 67. Note that only the first 54 characters of this label will be displayed and editable in the database. No other update should precede this transaction.

Pos.	Length	Value	Meaning
2	6	'X1HIST'	Line code for a session freeze
8	60		Comment visible on LH screen

For more details on the batch updating, refer to the corresponding chapter in the VisualAge Pacbase Interface User's Guide.

PRINTED OUTPUT

The two printed outputs generated by this procedure are:

- A global report on the update,
- A list of the rejected update transactions.

They are printed by the user, and the transaction groups are separated by a flag.

This procedure does not provide any generation or printing of data contained in the database. These are obtained via the Generation-Printing (GPRT) procedure.

RESULT

Output of the UPDT procedure is:

- A database ready to be used on-line or in batch mode.
- A Journal file of the transactions that have modified the database (as long as there was no inhibit request during the last restoration).

2.1.3. UPDT: DESCRIPTION OF STEPS

UPDT: 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 codes:
 -0:  OK.
 -4:  Database invalid, STOP triggered.


TRANSACTION FORMATTING:  PACA05

.Permanent input files:
 -Data file
  PAC7AR
 -Index File
  PAC7AN
 -Error message file
  PAC7AE

.Input transaction file:
 -Update transactions
  PAC7MB (MBUPDT file in INPUT directory)

.Output files:
 -Formatted transactions
  PAC7MV
  (must have capacity to contain all transactions in their
  complete state, plus the elementary delete transactions
  generated by the multiple delete transactions)
 -Work file
  PAC7MW
```

```
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
 - List of erroneous transactions
  PAC7IF
 (The list of transactions belonging to a user is preceded
 by a banner specifying the user code.)

.Return codes:
 - 0:  OK without error
 - 2:  Warning error
 - 4:  Serious error
```

2.1.4. UPDT: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) UPDT BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "--------------------------------------------------------"
echo "                        UPDT 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 : BATCH UPDATE
# ********************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7DS=$PACTMP'UPDTDS.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'MBUPDT'
 export PAC7MB
 PAC7MV=$PACTMP'MV'
 export PAC7MV
 PAC7MW=$PACTMP'MW'
 export PAC7MW
 echo "Execution : PACA05"
 rtscgi PACA05
 RETURN=$?
 case $RETURN in
 0)
  echo "Deletion of the temporary files"
  rm -f $PACTMP'MW'
  . $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
  . $PACDIR/assign/$1/SEMLOCK.ini
  PAC7MV=$PACTMP'MV'
  export PAC7MV
  PAC7IE=$PACTMP'UPDTIE.A15'
  export PAC7IE
  PAC7IF=$PACTMP'UPDTIF.A15'
  export PAC7IF
  echo "Execution : PACA15"
  rtscgi PACA15
  RETURN=$?
  case $RETURN in
  0)
   echo "End of procedure"
   echo ""
   echo "Deletion of the temporary files"
   rm -f $PACTMP'MV'
   ;;
  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 $RETURN
   ;;
  *)
   echo "Error in executing PACA15"
   ;;
  esac
  ;;
 *)
  echo "Error in executing PACA05"
  ;;
 esac
 ;;
4)
 echo "Error in executing PTUBAS"
 echo "Database unavailable"
 ;;
*)
 echo "Error in executing PTUBAS"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

2.2. UPDP: DATABASE UPDATE FROM PAF TABLES

2.2.1. UPDP: INTRODUCTION

UPDP: INTRODUCTION

The UPDP procedure performs an update of the network from a sequential file reflecting PAF tables.

The operating principle of UPDP is very similar to that of UPDT, with the exception that input transactions have a different format.

EXECUTION CONDITIONS

Refer to the 'EXECUTION CONDITIONS' section of the UPDT procedure.

ABNORMAL EXECUTION

Refer to the 'ABENDS' section of the UPDT procedure.

2.2.2. UPDP: INPUT - PROCESSING - RESULTS

UPDP: INPUT-PROCESSING-RESULTS

USER INPUT

The sequential file of input transactions is produced by a PAF extractor program. Its records mirror the PAF tables (described in the Pactables Manual).

Pos.	Length	Meaning
1	1	Transaction code (C, M, X, D or A, B)
2	10	PAF table code
12	299	PAF table contents (described in the Pactables Manual).

UPDATE RULES

Update transactions are not sorted.

Each set of transactions impacting a library or session must be preceded by an ASSIGN table code line.

Pos.	Length	Value	Meaning
2	10	'ASSIGN'	Table code
12	8	uuuuuuuu	User code
20	8	pppppppp	Password
28	3	bbb	Library code
31	4	ssss	Session number
		' '	current session
35	1	'T'	Session status: Test session
39	1	'A' or 'F'	Language code, useful if the transactions are not in the same language as the Database
			IN CASE OF A DSMS CONTROL OF THE DATABASE :
40	3	ppp	Product code
43	6	nnnnnn	Product number

When the update is performed while the conversational mode is active (on platforms that support this functionality), the input transaction flow must be preceded by a CHECKP table code line.

Pos.	Length	Value	Meaning
2	10	'CHECKP'	Table code
12	4	nnnn	Number of transactions processed between two pauses or checkpoints
16	4	'UPDT'	Update procedure
20	2	nn	OS/2, UNIX, WINDOWS NT: Pause time, in seconds, between two update sets

PRINTED OUTPUT

Refer to the description of the UPDT output.

RESULT

Refer to the description of the UPDT result.

2.2.3. UPDP: DESCRIPTION OF STEPS

UPDP: 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 codes:
 -0:  OK.
 -4:  Database invalid, STOP triggered.

TRANSACTION FORMATTING:  PAF900

.Permanent input files:
 -Data file
  PAC7AR
 -Index File
  PAC7AN
 -Error message file
  PAC7AE

.Input transaction file:
 -Update transactions
  PAC7GY

.Output files:
 -Formatted transactions
  PAC7MV
  (must have capacity to contain all transactions in their
  complete state, plus the elementary delete transactions
  generated by the multiple delete transactions)
 -Work file
  PAC7MW
```

```
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
 - List of erroneous transactions
  PAC7IF
 (The list of transactions belonging to a user is preceded
 by a banner specifying the user code.)

.Return codes:
 - 0:  OK without error
 - 2:  Warning error
 - 4:  Serious error
```

2.2.4. UPDP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) UPDP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        UPDP 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 : BATCH UPDATE FROM PAF TABLES
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7DS=$PACTMP'UPDPDS.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/$1/PAC7AR.ini
 PAC7GY=$PACINPUT'MBUPDP'
 export PAC7GY
 PAC7MV=$PACTMP'MV'
 export PAC7MV
 PAC7MW=$PACTMP'MW'
 export PAC7MW
 echo "Execution : PAF900"
 cobrun PAF900
 RETURN=$?
 case $RETURN in
 0)
  echo "Deletion of the temporary files"
  rm -f $PACTMP'MW'
  . $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
      . $PACDIR/assign/$1/SEMLOCK.ini
      PAC7MV=$PACTMP'MV'
      export PAC7MV
      PAC7IE=$PACTMP'UPDPIE.A15'
      export PAC7IE
      PAC7IF=$PACTMP'UPDPIF.A15'
      export PAC7IF
      echo "Execution : PACA15"
      rtscgi PACA15
      RETURN=$?
      case $RETURN in
      0)
       echo "End of procedure"
       echo ""
       echo "Deletion of the temporary files"
       rm -f $PACTMP'MV'
       ;;
      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 $RETURN
       ;;
      *)
       echo "Error in executing PACA15"
       ;;
      esac
      ;;
     *)
      echo "Error in executing PAF900"
      ;;
     esac
     ;;
    4)
     echo "Error in executing PTUBAS"
     echo "Database unavailable"
     ;;
    *)
     echo "Error in executing PTUBAS"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

2.3. GPRT: GENERATION AND PRINTING

2.3.1. GPRT: INTRODUCTION

GPRT: INTRODUCTION

The Generation and Printing procedure, GPRT, has a two-fold purpose:

. To print documentation using data contained in the database, and

. To generate Programs, Screens, Database descriptions, Data Structures, and error messages.

This procedure does not affect the database. Therefore, it may be executed while the files are open to on-line use.

However, if the on-line generation and print requests are to be included, then the Generation-Print Request (AG) file must be closed. (The procedure invalidates the printing requests entered on line, therefore the file must be accessible for update.)

It calls a unique program (PACBE), which is used as a monitor calling the different programs that make up the procedure.
All programs that make up the procedure are thus considered to be sub-programs of this monitor, with which they communicate by means of a communication area and certain return codes.

Since user requests are often diverse, this procedure is broken down into 'sub-chains' whose purpose is to process, in an integrated manner, the preparation of the generation-printing requests for the families they manage. They are identified by a one-position code as follows:

```
- A : Data elements
- B : Database blocks (DBD)
- C : COBOL programs (COB)
- D : Specifications Dictionary
- E : OLSD screens (OSD)
- G : Client/Server Screens (OCS)
- K : Error messages (OCS)
- L : Error messages (OSD)
- M : User manuals
- N : Personalized Documentation Manager (PDM)
- P : Batch programs (BSD)
- R : Production Environment Interface (PEI)
- Q : Relational-SQL Database blocks
- T : Windowing of OLSD applications
      (PAW, Pacbase Web Connection)
```

This code is referenced again in the names given to the programs, files and reports that are generated in this procedure. For programs, this is the fourth character of the code. Examples:

```
- PACA10 : General program.
- PACB30 : Database block extractor.
```

For files or reports, this is the last character of their external name. Examples:

```
- PAC7IA : General printing of command chain.
- PAC7GP : Generated file of batch programs.
```

Following the execution of the two general programs that are common to all chains (PACA10 and PACA20), the sub-chains are activated, if appropriate, in the following order:

```
- Production Environment Interface,
- Database Blocks,
- COBOL programs (COB),
- On-line Screens (OLSD),
- Client Screens,
- Server Screens,
- Error Messages and Dialog Windowing,
- Volumes,
- Personalized Documentation Manager,
- Batch programs,
- Specifications Dictionary.
```

Each sub-chain is structured in the same manner:

```
- The 'extraction' programs  (3x),
- The 'preparation' programs (4x),
- The 'generation' programs  (8x),
- The 'print' programs       (90).
```

These codes are found in the last two characters of the program codes of the procedure. Examples:

```
- PACB40 : Database block preparation,
- PACE80 : Screen generator.
```

Besides, a specific coding is used for file external names. It represents their use in the procedure:

```
- G : Generated code
- I : Reports
- J : Print requests
- K : Preparation for printing
- L : Error messages
- M : Transactions
- S : Skeletons
- W : Work
```

This code is found one character before last in the procedure files external name. Examples:

```
- PAC7GL : Generated error messages
- PAC7IN : Printing of Personalized Documentation
```

Files containing the 'generated source code' (ready to be compiled or to be stored in an Assembler or Source Library) are concatenated into a single physical file that will be used in the following step.

The Error Message file is updated using the file with an LG suffix, and is retrieved into the file with a GL suffix. The procedure does not include a name for the two versions of this file. Therefore, they must be specified when these messages are generated.

(The user error message file of the PAC700 6.2 type is retrieved into the file with a suffix of GM whose name must also be specified in a generation request.)

Volumes are standardly printed in an IN-suffixed file. The GN-suffixed file can also be used (record length = 265) with the 'ASA' skip character in the first position of each record when special print characteristics are needed.

The file containing the elements necessary for the windowing of OLSD applications is coded PAC7GT (record length is 180). Its name must be specified in the generation request.

EXECUTION CONDITIONS

The files can remain open, except if the generation-print of on-line requests was requested via the ' +AG' command. In this case, the Generation-Printing Request file (AG) must be closed.

ABNORMAL EXECUTION

Refer to chapter 'OVERVIEW', subchapter 'Abnormal Endings' in the 'Batch procedures Manual: the Administrator's Guide'.

2.3.2. GPRT: STRUCTURE OF REQUESTS

GPRT: STRUCTURE OF THE REQUESTS

The GPRT requests are structured in three parts:

. The actual request, coded in a way similar to on-line selection,

. a report formatting option, coded in a way similar to the operation code,

. the code of the entity concerned, if relevant

In some cases, parameters may be necessary. Parameters can be specified in two places :

. in pre-formatted fields, with the input of the command code on GP screen,

. on a continuation line, by placing the asterisk ('*') in the continuation line field
(continuation of command label on batch form Z).

Presentation options and all possible parameters are indicated for each GPRT command in section 'Generation/Printing commands' as well as for each entity in the corresponding manual.

STRUCTURE OF THE COMMAND

The Generation/Printing request of an entity breaks down in three parts :

The first part indicates the nature of the generation/printing:

. L : List of entities,

. D : Description of entities,

. G : Generation (of programs, screens, database Blocks, error messages..).

. P : Printout (User manuals or reports).

The second part specifies the printing criteria for example for lists, the ordering criteria (by code, by name, by type...).

The third part gives the name of the entity :

for a methodology entity, the (M) type is completed to specify if it is a Property (P), an Object (O), a Relation (R) or a Functional Integrity Constraint (C).

SPECIAL COMMANDS

. FLx : Flow control cards (x = type of entity) flow of compilations following the generation.

. JCL : Allows the user to set up JCL lines for the on-line GPRT start-up (see section 'Generation/ Printing commands' hereafter).

. UPC : transformation of lowercase characters into uppercase characters for printers which do not support lowercases.

To consult the complete list of the commands and their meaning, see section 'Generation/Printing commands' hereafter.

PRINTING BY KEYWORD

To obtain a printout by keyword, enter a 'K' as the second character of the command. In this case, after the line has been created, a 'continuation' line is automatically displayed. The user can enter on this line the keyword(s) for which a printout is requested.

Furthermore, the print name contains a selection field in which the user can specify whether the selection is to be made:

. On the whole set of keywords (SPACE),

. On the keywords automatically derived from the name (L),

. On explicit keywords (M).

2.3.3. GPRT: GENERATION/PRINTING COMMANDS

```
--------------------------------------------------------------------------------
! VA Pac                APPLICATION DEVELOPMENT        SG000008.LILI.CIV.1583  !
!GENERATION AND PRINT COMMANDS                             USER: 21 SG000008  !
!1 2  3   4        56 7 8 <-------------------- 9 AND 10 ------------------>   !
!A SO COM ENTITY : OP V C CONTINUATION OF REQUEST                              !
!     LKP        : C1   * LIST OF PROGRAMS RELATED BY KEYWORDS        SEL:_    !
!                :        ----------------------------------------------- 17   !
!     UPC        : C1     SHIFT TO UPPERCASE    MANUAL:_ DOC:_ ERROR MESS:_    !
!                :                                   13    14            15    !
!  90 FLP        : C1     PROGRAM JOB CARD / JOB DELIM  ENV: _ (CCF:_ CCB:_)   !
!                :                                                13    15     !
!  90 GCP PA10FL : C1     SOURCE CODE FOR SELECTED PROGRAM     (CCF:_ CCB:_)   !
!                :                                                13    15     !
!  90 GCP PA20PA : C1     SOURCE CODE FOR SELECTED PROGRAM     (CCF:_ CCB:_)   !
!  90 GCP PA30AR : C1     SOURCE CODE FOR SELECTED PROGRAM     (CCF:_ CCB:_)   !
!  91 FLO        : C1     SCREEN JOB CARD / JOB DELIM   ENV: _ (CCF:_ CCB:_)   !
!  91 GCO DO0000 : C1     SCREEN'S PGM AND MAP SOURCE CODE   (CCF:__ CCB:__)   !
!                :                                            13-14   15-16    !
!  96 PCV VOLUME : C1     PRINT VOLUMES BY CHAP / SUBCHAP AND CODE: _ __ __    !
!                :                                               18 19 20      !
!                :                                                             !
!*** END ***                                                                   !
!                                            11                                !
!O: C1 CH: GP                          JOB:          PASSWORD:                 !
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
! VA Pac                APPLICATION DEVELOPMENT        SG000008.LILI.CIV.1583  !
!VALID GENERATION AND PRINT COMMANDS                        USER: SG000008     !
!                                                                              !
!A SO COM ENTITY : OP   C CONTINUATION OF REQUEST               : LIB SESSI    !
!     JCL 000000 :        //PSTSG8 JOB (634,CGI46808),SG8,CLASS= :             !
!     JCL 000020 :        //   EXEC ZA73GPRT,ROOT=LI,FILE=LI,OUT :             !
!     JCL 000030 :        //   LOADTP='PST.CICS.LINKLIB',OUTL=R, :             !
!     JCL 000040 :        //   INDUV='PST',INDSV='PST',INDSN='PS :             !
!     JCL 000045 :        //   STEPLIB='PST.PAC73.MBR7',         :             !
!     JCL 000050 :        //   LOADBA='PST.BATCH.LINKLIB'        :             !
!     JCL 600100 :        //PAC.PAC7SC DD DSN=PST.LILISCA,DISP=S :             !
!     JCL 600200 :        //PAC.PAC7SG DD DSN=PST.LILISGA,DISP=S :             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!UPDATE INHIBITED WITH THIS DISPLAY TYPE                                       !
!O: C2 CH: GP                          JOB:          PASSWORD:                 !
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
! VA Pac                APPLICATION DEVELOPMENT        SG000008.LILI.CIV.1583  !
!GENERATION AND PRINT COMMANDS                              USER: SG000008     !
!                                                                              !
!A SO COM ENTITY : OP V C CONTINUATION OF REQUEST                : LIB SESSI   !
!     JCL 000000 :    V   //PSTSG8 JOB (634,CGI46808),SG8,CLASS= :             !
!     JCL 000020 :    V   //   EXEC ZA73GPRT,ROOT=LI,FILE=LI,OUT :             !
!     JCL 000030 :    V   //   LOADTP='PST.CICS.LINKLIB',OUTL=R, :             !
!     JCL 000040 :    V   //   INDUV='PST',INDSV='PST',INDSN='PS :             !
!     JCL 000045 :    V   //   STEPLIB='PST.PAC73.MBR7',         :             !
!     JCL 000050 :    V   //   LOADBA='PST.BATCH.LINKLIB'        :             !
!     JCL 600100 :    V   //PAC.PAC7SC DD DSN=PST.LILISCA,DISP=S :             !
!     JCL 600200 :    V   //PAC.PAC7SG DD DSN=PST.LILISGA,DISP=S :             !
!  90 FLP        : C1     PROGRAM JOB CARD / JOB DELIM  ENV: _ ( : ITF         !
!  90 GCP PA10FL : C1     SOURCE CODE FOR SELECTED PROGRAM     ( : ITF         !
!  90 GCP PA20PA : C1     SOURCE CODE FOR SELECTED PROGRAM     ( : ITF         !
!  90 GCP PA30AR : C1     SOURCE CODE FOR SELECTED PROGRAM     ( : ITF         !
!  91 FLO        : C1     SCREEN JOB CARD / JOB DELIM   ENV: _ ( : ITF         !
!  91 GCO DO0000 : C1     SCREEN'S PGM AND MAP SOURCE CODE   (CC : ITF         !
!  91 FLO        : C1     SCREEN JOB CARD / JOB DELIM   ENV: V ( : SG8         !
!  91 GCO PA0030 : C1     SCREEN'S PGM AND MAP SOURCE CODE   (CC : SG8         !
!  91 GCO PA8888 : C1     SCREEN'S PGM AND MAP SOURCE CODE   (CC : SG8         !
!                :                                                             !
!UPDATE INHIBITED WITH THIS DISPLAY TYPE                                       !
!O: C3 CH: GP                          JOB:           PASSWORD:                !
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
! VA Pac                APPLICATION DEVELOPMENT        SG000008.LILI.CIV.1583  !
!JCL LINES FOR THE COMMANDS                                 USER: SG000008     !
!                                                                              !
!A   COM  LINE   :    V C CONTINUATION OF REQUEST                              !
!    JCL  000000 :    V   //PSTSG8 JOB (634,CGI46808),SG8,CLASS=X,MSGCLASS=C   !
!    JCL  000020 :    V   //   EXEC ZA73GPRT,ROOT=LI,FILE=LI,OUTL=R,OUT=C,     !
!    JCL  000030 :    V   //   LOADTP='PST.CICS.LINKLIB',OUTL=R,UTI='SG8',     !
!    JCL  000040 :    V   //   INDUV='PST',INDSV='PST',INDSN='PST',            !
!    JCL  000045 :    V   //   STEPLIB='PST.PAC73.MBR7',                       !
!    JCL  000050 :    V   //   LOADBA='PST.BATCH.LINKLIB'                      !
!    JCL  600100 :    V   //PAC.PAC7SC DD DSN=PST.LILISCA,DISP=SHR             !
!    JCL  600200 :    V   //PAC.PAC7SG DD DSN=PST.LILISGA,DISP=SHR             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!                :                                                             !
!*** END ***                                                                   !
!O: C4 CH: GP                          JOB:           PASSWORD:                !
--------------------------------------------------------------------------------
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1		ACTION CODE
2	2		PROCESSING SEQUENCE ORDER This field is used to specify the sequence in which print requests are processed and printed. If no value is entered in this field, the print requests are processed according to their position in the input sequence displayed on the screen. If an alphanumeric value is entered, reports are printed and sorted on this value basis. In case of generation request, this criterion is forced automatically by the system in order to sort the generations by entity types:
		90	Programs
		91	Screens
		92	Database Blocks
		93	User Manuals
		94	Error messages
		95	Data structures
		96	Volumes (PDM)
			The request criteria for a modification of the flow nanagement is also forced according to the entity to generate. The ordering criteria assigned automatically cannot be modified by the user. If the user tries to modify these values, the system retrieves automatically the values specified above without issuing an error message.
3	4		GENERATION-PRINT COMMAND NOTE: Input of the entity code is required or optional depending on the command. The following indicators describe the various options: (A) Required occurrence code input (Batch column 9). (B) Optional occurrence code input. If omitted, all occurrences of the entity type are listed in the user's hierarchical view. (C) Occurrence code input not allowed. All occurrences of the entity type are listed in the user's hierarchical view. (D) A blank line may be requested by placing an asterisk in the CONTINUATION OF REQUEST INDICATOR(C)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			field and pressing the ENTER key. What may be entered on this line depends on the command; you will find below what options are possible. This corresponds to batch columns 31 to 80 incl. NOTE: Each command has different requirements with respect to the type of additional information to be supplied. Values may be entered here, or left blank for the default. The following list identifies by code the information expected for each command: (1) SEL: _ Limit the list by keyword type. Enter 'M' for explicit, 'L' for implicit, or blank for both. In batch mode, enter this value in column 30. See also SELECTION OF KEYWORD TYPE. (2) Same as above plus a following line on which a user may enter one or several keywords. This appears as a continuation line in on-line mode, and corresponds to batch columns 31 to 80. (3) FORMAT: _ A format may be specified by entering 'I' for internal, 'E' for input, or 'S' for output. Enter these values in column 17 in batch mode - a blank is also valid and means that the default value is desired. See also TYPE TO SELECT. (4) CCF:_ CCB:_ The code of the control card in front of program and in back of program, respectively. Enter these codes in columns 19 to 22 in batch mode. The codes must be consistent with the codes displayed on the Dialogue Definition screen. (5) CCF:__ CCB:__ The code of the control card in front of program and in front of map, and the code of the control card in back of program and in back of map, respectively. The user can override the default control cards. These codes should be consistent with the values on the Dialogue Definition. In batch mode, use columns 19 to 22. (6) TYPE: __ The user enters the selected type which should be consistent with the corresponding field on the definition screen of that entity type. In batch mode enter the type in columns 17 and 18.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			(7) PRINT VOLUME BY CHAP/SUBCHAP AND CODE:_ __ __ Specify the chapter and/or subchapter. Enter 'C' for chapter followed by the chapter code, or 'S' for subchapter followed by the chapter and subchapter codes. In batch mode use columns 23 through 27. (8) ENV.:__ (CCF:__ CCB:__) For those sites that are using the PEI option: the environment may be specified. In batch mode enter the environment code in column 17, and the corresponding control cards in columns 19 through 22. THESAURUS ---------
		DCK	(C) A complete description of keywords defined in the thesaurus which lists the SYNONYM OR DEFINITION field contents associated with each keyword. NOTE: This data being specified in Inter-Library only, this command cannot be used with the U1 option. Use the C1 or I1 option which gives the same output.
		LCK	(1) (C) A listing of all keywords defined in the thesaurus, with their synonyms. It includes the number of uses of these keywords in the Database. TEXTS -----
		DCT	(A) Description of selected Text. NOTE: If you enter an "*" in the ENTITY CODE field, descriptions of all Text occurrences will be printed, sorted by code.
		DTT	(B) (6) Descriptions of Text occurrences, sorted by type.
		L*T	List of Texts with their paragraphs titles, sorted by code.
		LCT	(C) List of Text occurrences, sorted by code.
		LKT	(2) List of Text occurrences whose names and/or explicit Keywords contain the Keyword(s) specified.

NUM	LEN	CLASS	DESCRIPTION OF FIELDS
		VALUE	AND FILLING MODE
		LTT	(6) List of Text occurrences, sorted by type.
			DOCUMENTS (PDM) ---------------
		FLV	(C) (D) (4) This command is used to specify the job card and end-of-job delimiters: Flow control for Documents. Use the continuation line to define user parameters on the control cards.
		LCV	(C) List of Documents, sequenced by code.
		LKV	(C) (2) List of Documents selected according to the keyword(s) entered on the continuation line.
		DCV	(B) Printing of the description of the Document whose code is entered in the Entity field. When this code is not entered, the descriptions of all the Documents are printed, sequenced by code.
		PCV	(B) (D) (7) Printing of the contents of the Document whose code is entered in the Entity field. When this code is not entered, the contents of all the Documents are printed, sequenced by code. For local printing in RTF format, the Document must be generated with the C2 option. Partial printing is documented in the 'Personalized Documentation Manager' Reference Manual, Chapter 'Access Commands', Subchapter 'Generation-Print'.
			ELEMENTS AND PROPERTIES ----------------------
		DCE	(B) A complete description of the defined element(s). The information is sequenced by element code. To get assigned text, use print option "2".
		DFE	(B) A listing of the element(s) not defined in the Specifications Dictionary, with cross-references.
		LACE	(C) A list of elements and properties, by Cobol name.
		LCE	(B)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			A list of defined elements, sequenced by element code.
		LKE	(C) (2) A list of elements and properties by keyword.
		LNE	(C) A list of elements and properties sequenced by element name.
		LXE	(C) A list of defined elements and properties which are not used.
			DATA STRUCTURES ---------------
		DCD	(B) A complete description of the data structure(s). This includes cross-references to programs and screens and a list of associated reports and segments. The information is sequenced by data structure code. Note: To get the associated text use print option "2".
		FLD	(C) (D) (4) This command is used to specify the job card and end-of-job delimiters: flow control of data structures. Use the continuation line to define user parameters on the control cards.
		GCD	(A) Generate a COBOL description (COPY book) of the data structure. For more details concerning generation, refer to the chapter corresponding to the 'DICTIONARY' reference manual.
		LCD	(C) A list of data structures sequenced by data structure Code.
		LTD	(C) A list of data structures sequenced by data structure type.
		LPD	(C) A list of data structures sequenced by external name.
		LKD	(C) (2) A list of the data structures whose names and/or explicit keywords contain the keyword(s) specified.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			SEGMENTS AND LOGICAL VIEWS -------------------------
		LCS	(C) List of Segments sorted by Code.
		LKS	(C) (2) List of Segments whose names and/or explicit keywords contain the keyword(s) specified.
		DCS	(B) (D: when entity code has been entered) (3) NOTE: Enter the Data Structure code in the ENTITY CODE field, and the Segment code(s) on the continuation line(s). A complete description of the Segment(s). This includes cross-references to Programs and Screens for the Data Structure and to all entities for the Segment(s) and a list of associated Reports and Segments. For Segments defined as tables with the Pactables function, a list of sub-schemas and sub-systems is printed. NOTE: To get the associated text for both the Segment and the Data Structure, use print option "2".
			INPUT AIDS ----------
		DCI	(C) A complete description of the input aid(s) including a list of uses of the input aid(s) in other entities. The information is sequenced by PIA code.
		LCI	(C) A list of input aids sequenced by the PIA code.
		LKI	(C) (2) A list of the input aids whose names and/or explicit keywords contain the keyword(s) specified.
		LXI	(C) List of all Cross-References (PIA Calls) as defined on the PIA description screen sequenced by the value of this field.
			DATABASE BLOCKS ---------------
		DTB	(B) (6) Description(s) of database blocks of the type specified including cross-references to other blocks and

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			screens. Note: To get the associated text, use print option "2"
		FLB	(C) (D) (4) (8) This command is used to specify the job card and end-of-job delimiters: Flow control of the block.
		FLS	(C) (D) (4) (8) Same as FLB for Relational/SQL blocks. Use the continuation line to define user parameters on the control cards.
		GCB	(A) (D) (4) Generate a DDL description of the database block specified (including 'DB'-type blocks for DB2). Use the continuation line to define the user parameters on the control cards.
		GSQ	(A) (D) (4) Generates the SQL DDL for the Relational/SQL database block specified. Use the continuation line to define the user parameters on the control cards.
		LCB	(C) List of database blocks sequenced by block code.
		LEB	(C) List of database blocks sequenced by external name.
		LKB	(C) (2) A list of the database blocks whose names and/or explicit keywords contain the keyword(s) specified.
		LTB	(C) (6) A list of database blocks whose block types have been defined with the specified value.
		LTS	(C) A list of SQL objects sequenced by code.
		LES	(C) List of SQL objects sequenced by external name.
			BUSINESS COMPONENTS, FOLDERS, FOLDER VIEWS, C/S SCREENS, SCREENS, DIALOGUES --
		DCO	(A) Complete Screen Description including Dialogue Complement and uses in other Screens. For Screens, information is also provided on relevant

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			Segments, Macro-structure calls, Beginning insertions modifications, Work Areas and Structured Code. NOTE: To get the associated text, use print option "2"
		FLO	(C) (D) (4) (8) This command is used to specify the job card and end-of-job delimiters: Flow control for screens. Use the continuation line to define user parameters on the control cards.
		GCO	(A) (D) (5) Generate a COBOL description of the Screen specified. Use the continuation line to define user parameters on the control cards.
		LCO	(C) List sorted by code.
		LNO	(C) List sorted by type.
		LPO	(C) List sorted by external program name.
		LSO	(C) List of (C/S) Screens sorted by external map name.
		LKO	(C) (2) List of occurrences whose names and/or explicit keywords contain the keyword(s) specified.
		LTO	(C) List of Screens sequenced by transaction code.
		DGC	(A) A complete description of a C/S Screen.
		DGS	(A) A complete description of a Business Component.
		GGC	(A) (D) (5) Generate a C/S Screen (TUI Client component).
		GGS	(A) (D) (5) Generation applicable to Business Component, Communication Monitor, Error Server, Folder.
		GVC	(A) (D) (5) Extract a Proxy object. Applicable to Folder View, Folder, Business Component.
		FGC	(C) (D) (4) (8) This command is used to specify the job card and end-

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			of-job delimiters: Flow control for C/S Screen.
		FGS	(C) (D) (4) (8) This command is used to specify the job card and end-of-job delimiters: Flow control.
			REPORTS -------
		DCR	(B) (D: when the entity code has been entered) NOTE: When requesting the description of a single Report, enter the Data Structure code in the ENTITY CODE field and the last character of the Report code on the continuation line. A complete description of the Report(s). This includes Report layouts. The information is sequenced by the Report code. Note: To get the associated text, use print option "2"
		LCR	(C) List of Reports sequenced by Report Code.
		LTR	(C) List of Reports sequenced by Type.
		LKR	(2) A list of the Reports whose names and/or explicit keywords contain the keyword(s) specified.
			PROGRAMS --------
		DCP	(B) A complete description of Program(s). The information is sequenced by the Program code. NOTE: To get the associated text, use print option "2"
		FLP	(C) (D) (4) (8) This command is used to specify the job card and end-of-job delimiters: Flow control for Programs. Use the continuation line to define user parameters on the control cards.
		FSP	(C) (D) (4) (8) This command is used to specify the job card and end-of-job delimiters: Flow control for "reverse engineered" programs. Use the continuation line to define user parameters on the control cards.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		GCP	(A) (D) (4) Generate a COBOL description of the Program specified. Use the continuation line to define user parameters on the control cards.
		GSP	(A) (D) (4) Generate a COBOL description of the "reverse engineered" Program specified. Use the continuation line to define user parameters on the control cards.
		LCP	(C) List of Programs sequenced by program code. Note: To get keywords, use print option "2".
		LTP	(C) List of Programs sequenced by type.
		LEP	(C) List of Programs sequenced by external name.
		LKP	(2) A list of the Programs whose names and/or explicit keywords contain the keyword(s) specified.
		DSP	(S) Description of the selected Program produced by REVERSE ENGINEERING.
			METHOD ENTITIES ---------------
		DCM	(A) A complete description of the Method entity as specified.
		DCMC	(C) A complete description of Method Functional Integrity Constraint(s).
		DCMO	(C) A complete description of Method Object(s).
		DCMR	(C) A complete description of Method Relationship(s).
		LCMC	(C) List of Method Functional Integrity Constraints sequenced by F.I.C. code.
		LCMO	(C) List of Method Objects sequenced by Object code.
		LCMP	(C)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			List of properties sequenced by Property code.
		LCMR	(C) List of Method Relationships with their Functional Integrity Constraints, sequenced by Relationship code.
		LKM	(C) (2) A list of the Method entities whose names and/or explicit keywords contain the keyword(s) specified.
			USER ENTITIES -------------
		DCF	(B) A complete description of the User Entity(s). The information is sequenced by User Entity code.
		DCQ	(B) A complete description of the User-Defined Relationship. The information is sequenced by Relationship code.
		DC$	(B) A complete description of the User Entity Occurrence(s). The information is sequenced by user entity type code.
		LCF	(C) List of User Entities sequenced by code.
		LCQ	(C) List of User-Defined Relationships sequenced by code.
		LC$	(C) List of User Entity Occurrences sequenced by User Entity type code.
		LK$	(2) (A) A list of the User Entity Occurrences whose names and/or explicit keywords contain the keyword(s) specified.
		LKF	(2) (C) A list of the User Entities whose names and/or explicit keywords contain the keyword(s) specified.
		LKQ	(2) (C) A list of the User-Defined Relationships whose names and/or explicit keywords contain the keyword(s) specified.
			NOTE ---- For all printing by keyword, you can specify the type

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			of selection (BLANK, L or M) on the print line. Keywords are indicated on the continuation line sent back by VisualAge Pacbase.
			ERROR MESSAGES -------------
		FLE	(C) (D) (4) This command is used to specify the job card and end-of-job delimiters: Flow control for error messages. Use the continuation line to define user parameters on the control cards.
		LEC	(A) List the error messages defined for the client component and for each client screen. This list only includes messages that have already been generated.
		LED	(A) List the error messages defined for the data structure and for each segment. This list only includes messages that have already been generated.
		LEO	(A) List the error messages defined for the dialogue and for each screen. This list only includes messages that have already been generated.
		GEC	(A) (D) Pacbench C/S: C1 : Error messages defined for the Client or Server Dialog and for each component. C2 : Error messages generated through option 1 plus documentary help messages. C3 : Error messages defined for the Client Dialog only.
		GED	(A) (D) C1 : Error messages generated for a Data Dtructure and for each Segment. C2 : Error messages generated through option 1 plus documentary help messages.
		GEO	(A) (D) OLSD Function: C1 : Error messages defined for the Dialog and for each Screen. C2 : Error messages generated through option 1 plus documentary help messages. C3 : Error messages for the Dialog only. C4 : Creation of the file required by Pacbase Web Connection. This command is applicable to the Dialogue.
			NOTE:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			If a Segment/Screen suffix is entered on the continuation line of one of the four preceding commands, error error messages are generated/printed only for the selected Segment/Screen.
			JCL INTRODUCTION ----------------
		JCL	This indicates that the COMMAND LABEL/SYSTEM RESPONSE field will contain JCL. The JCL command can only be entered in the 'C4' screen format option.
			SHIFT TO UPPERCASE ------------------
		UPC	This command allows for the automatic transformation of lowercase into uppercase in the printed output of the GPRT procedure. When the UPC command is entered, the following line is displayed: SHIFT TO UPPERCASE MANUAL:_ DOC:_ ERROR MESS:_ The PACBASE user must specify to which type of GPRT output the UPC command will apply (even when only one GPRT command is validated). In order to do this, the value '1' must be entered in one of the three fields displayed above: in the MANUAL field for User Manuals (U) or Volumes (V); in the DOC field for entity-related commands; in the ERROR MESS field for the generation of error messages. NOTE: This also allows for the selective implementation of the UPC command when the execution of several GPRT jobs is requested and the SHIFT TO UPPERCASE must not apply to all of them, in which case the corresponding field(s) must be left blank.
			JOB STREAM CARDS ----------------
		FGC	Stream check: C/S screen
		FGS	Stream check: Business Component
		FLO	Stream check: Screens
		FLS	Stream check: SQL relational Database Blocks

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		FLB	Stream check: Database Blocks
		FLD	Stream check: Data Structures
		FLP	Stream check: Programs
		FSP	Stream check: Programs from REVERSE ENGINEERING
		FLV	Stream check: Report
		FLE	Stream check: Error Messages
			PAF TABLES OF METHODOLOGY-SPECIFIC ENTITIES --
		PCM	Description of PAF Tables for entities specific to a methodology. This command necessarily followed by a Methodology code (see next field).
4	6		ENTITY CODE This field is displayed with the label 'ENTITY' on screen format options '1', '2' and '3' of the GP screen. When required, the user enters the entity code which corresponds to the COMMAND FOR PRINT REQUEST. 'PCM' COMMAND: You enter in this field the code of the selected Methodology:
		M	Merise
		D	YSM
		A	SSADM
		O	OMT
		F	IFW
			'JCL' COMMAND: The JCL lines will be sorted according to the number entered in this field. On the screen format option '4' of the GP screen, this field is displayed with the label 'LINE'.
		<600000	JCL lines at the beginning of the job stream.
		>599999	JCL lines at the end of the job stream.
			OPERATION CODE
5	1		LIBRARY VIEW SELECTION CODE Used to select the libraries from which the entities are to be generated and/or printed.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			This code has the same meaning as the first character of the OPERATION CODE field on all VisualAge Pacbase screens.
		C	Default value: Selected library and higher level libraries. In case of duplicates, the lines from the lower level library are taken into account. NOTE: IN GENERATION THE VALUE 'C' IS AUTOMATICALLY ASSIGNED BY THE SYSTEM.
		I	Selected library and lower and higher level libraries.
		U	Selected library only.
		A	Selected library and higher level libraries with display of duplicates.
		>	Higher level libraries only.
		<	Lower level libraries only.
		Z	Selected library and lower level libraries.
6	1		PRINT OPTION This field does not appear on the "C4" screen format option. Used to indicate that sub-reports be included.
		1	Default
		2	Add Associated Text to the output, depending upon the value entered in the COMMAND FOR PRINT REQUEST. See the specific Command for Print Request.
7	1		VALIDATION OF COMMAND REQUEST This field does not appear on the "C2" screen format option.
		blank	The value in the COMMAND FOR PRINT REQUEST field is not to be taken into account.
		V	The COMMAND FOR PRINT REQUEST is validated. NOTE: These commands must be re-validated each time a request is made.
8	1		CONTINUATION OF REQUEST INDICATOR
		blank	No continuation line is requested.
		*	A continuation line is requested (or displayed) for this GP command.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			For some specific Generation-Print Requests, this field is automatically filled by VA Pac (for instance a Request by Keywords). You must then fill in the continuation line's input fields. NOTE: A maximum of 5 continuation lines is authorized.
9	50		COMMAND LABEL / SYSTEM RESPONSE This field has three functions: - With screen format option "1", the system uses this field to display a system response line which is the label for the COMMAND FOR PRINT REQUEST entered. - With certain commands the user is asked to enter additional information. Also see the SYSTEM RESPONSE REQUEST and CONTINUATION LINE fields. - With the 'C4' screen format option, the user can enter JCL lines, which will or will not be taken into account, depending on the value entered in the VALIDATION OF COMMAND REQUEST field.
10	50		CONTINUATION LINE This line is displayed on-line. It represents columns 31 through 80 on Batch Form 'Z'. This line serves many purposes, among them: . To specify keywords. See COMMAND FOR PRINT REQUEST field, note (2). . To generate error messages of one screen, the Dialogue code is entered in the ENTITY CODE field and the screen suffix in the CONTINUATION LINE field.
11	3		JOB SUBMISSION REQUEST Used to automatically submit the generation and/or printing job from the GP screen when the operating system and TP monitor in use allow for this. The job stream will contain only validated commands for generation and/or print requests and validated JCL lines, all libraries and sessions included.
		blank	No job submission. Update the AG file.
		JOB	Job submission. NOTE: For IMS, system messages are displayed. See USER'S MANUAL, chapter "CHOICE: ACCESS COMMANDS", subchapter "SPECIAL CHOICES: IMS VERSION".
		SUB	Job submission.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			NOTE: For IMS, system messages are not displayed.
			SYSTEM RESPONSE REQUEST The following fields appear in the COMMAND LABEL/ SYSTEM RESPONSE field only on the 'C4' screen format option for certain Commands for Print Request. They prompt the user for additional input depending on the command entered.
12	2		TYPE TO SELECT A. TYPE TO SELECT (2-character field): Used to specify an occurrence type when requesting a List or Description sorted by type. B. FORMAT TO SELECT (1-character field): Used to specify the Segment format when entering a DCS command.
		blank or C	Printing of data related to validations and updates performed by user programs on the Segment's Data Elements. In addition, internal and input formats are printed.
		E	Input format only.
		I	Internal format only.
		R	Validations, updates, relational names.
		S	Output format only.
13	1		CARDS IN FRONT PGM/UPPERCASE SHIFT GENERATION ---------- Enter the one-character code that identifies the job card to be inserted before the generated occurrence. Default: Code entered in the Library Definition. NOTE: This value may be overridden on the occurrence's Definition. Also see Subchapter "OPTIONAL CONTROL CARDS UPDATING", Chapter "DATABASE MANAGEMENT", OPTION CODE field in the VA Pac TUI User Interface Guide (Ref. DD USE). SHIFT TO UPPERCASE FOR VOLUMES ----------------------------- Volumes ('V' entity) are printed in uppercase characters with the UPC command.
		1	YES.
		0	NO (Default option).
14	1		CARDS IN FRONT MAP/UPPERCASE SHIFT

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			SCREEN GENERATION ----------------
			The one-character code that identifies the job card to be inserted before each generated screen map. This code is entered on the Dialogue or Screen Definition screen and may be overridden here. Also see: OPTION CODE and INPUT PARAMETERS fields in the 'OPTIONAL CONTROL CARDS UPDATING' Subchapter, 'DATABASE MANAGEMENT' Chapter in the VA Pac TUI User Interface Guide (Ref. DD USE).
		$	No generation of map. (Use this value in conjunction with the CONTROL CARDS IN BACK OF MAP field.)
			SHIFT TO UPPERCASE FOR LIST/DESCRIPTION PRINT OUTPUT --
			Print output shifted to uppercase with UPC command.
		1	YES.
		0	NO (Default option).
15	1		CARDS IN BACK / UPPERCASE SHIFT
			GENERATION ----------
			Enter the one-character code that identifies the job card to be inserted after the generated occurrence.
			Default: Code entered on the Library Definition.
			NOTE: This value may be overridden on the occurrence Definition.
			SHIFT TO UPPERCASE FOR PRINTED ERROR MESSAGES ---
			Error messages are printed in uppercase characters with the UPC command.
		1	YES.
		0	NO (Default option).
16	1		CONTROL CARDS AFTER MAP
			Screen and C/S Screen entities:
			The one-character code that identifies the job card to be inserted after each generated Screen or Screen c/s map.
		$	No generation of map.
			NOTE: This field is not used in a Pacbench C/S

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			development with the specification of Folder. Business Component / single-view (with no specification of Folder): Option code which selects the JCL lines to be inserted after the Services Manager generated.
17	1		SELECTION OF KEYWORD TYPE
		blank	Selection on both implicit and explicit keywords.
		L	Selection on implicit keywords only.
		M	Selection on explicit keywords only.
18	1		DOCUMENT SELECTIVE PRINT REQUEST Field displayed with PCV command only. NOTE: Applicable only when the Volume has a Chapter/ Subchapter Description Organization Mode (Value '0' in corresponding field in Volume Definition).
		blank	Print the whole Volume (default value)
		C	Print the selected chapter (see next field)
		S	Print the selected subchapter (see next two fields)
19	2		CODE OF THE CHAPTER TO BE PRINTED Field displayed with PCV command only. Code of the chapter to be printed, or the chapter that contains the subchapter to be printed.
20	2		CODE OF THE SUBCHAPTER TO BE PRINTED Field displayed with PCV command only. Code of the subchapter to be printed.
21	8		CODE OF RECIPIENT USER FOR JCL COPY This field is reserved for on-line use. If you have a 4-level authorization, this field allows you to initialize another user's JCL lines. To do so, when the JCL lines are displayed, override your user code with that of the other user. Press the ENTER key.

CODING OF GPRT OUTPUT FILES CREATED ON DISK

All output files generated by the GPRT procedure are created in the Temporary Files sub-directory (procedure 3rd parameter).

These files follow a special codification in order for the user to find his/her generated programs or reports easily.

GENERATED SOURCE AND PRINT FILES:

These files are assigned the "GPRT." prefix:

GENERATED SOURCE	PRINT FILES
GPRT.GB (Database Blocks)	GPRT.IA (Report)
GPRT.GD (Data)	GPRT.ID (Data)
GPRT.GE (Screens - OSD)	GPRT.IH (PEI)
GPRT.GP (Programs)	GPRT.IL (OSD Error Mes.)
GPRT.GQ (SQL)	GPRT.IM (User Manuals)
GPRT.GR (Reverse)	GPRT.IN (PDM-Volumes)
GPRT.GG (Client screens)	GPRT.IK (OCS Error Mes.)
GPRT.GV (Server screens)	GPRT.II (ICS Generat. Err)

ERROR MESSAGE FILES:

These files are assigned the "ERR." prefix:

```
      Input files : ERR.LG (OSD) and ERR.LK (OCS)
      Output files: ERR.GL (OSD) and ERR.GK (OCS)
```

At the end of the procedure, a COPY order ensures the rotation from GL to LG and GK to LK.

ON-LINE APPLICATIONS AUTOMATIC REVAMPING FILE:

This file is assigned the "PAW." prefix:

PAW.GT contains the necessary elements for windowing.

TEMPORARY FILES:

There are two types of temporary files:

. Files internal to the GPRT procedure:

These files are assigned the "W" prefix and are deleted at the end of the procedure.

. Files which may be of interest to the VisalAge Pacbase user:

These files are assigned the "X" prefix. They are deleted at the end of the procedure unless the corresponding line found at the end of the procedure is inhibited by entering REM on the DELETE line which corresponds to the file to be retrieved.

These files are:

```
XGI (Va Pac-GIP Interface)
XGM (PAC700-type labels)
XGN (Volumes on 265 characters).
```

EXECUTION CONDITION

On-line Servers may be operational.

ABENDS

Refer to Chapter "DESCRIPTION OF BATCH PROCEDURES", Subchapter "ABENDS".

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

NOTE CONCERNING THE GENERATION OF ERROR MESSAGES

It is advisable to request the generation of Error Messages (GEO or GCO command) in batch mode rather than using the Generation & Print Commands screen (CH: GP).

The Batch Server, which processes the Generation-Print requests submitted from the "GP" screen, does not perform the rotation of the generated sequential files; therefore there can be no cumulative generation.

As a result, error messages generated in prior on-line requests are lost.

In order to avoid this problem, the indexed Error Message file must be routinely loaded via the EMUP procedure after each sequential file generation.

As a default, the GPRT procedure does not perform a cumulative generation of error messages, the LG and LK files being assigned as null files.

To activate the cumulative generation, assign the files as follows:

```
PAC7LG=$PACTMP'ERR.LG'
PAC7LK=$PACTMP'ERR.LK'
```

2.3.4. GPRT: USER INPUT AND RESULTS

GPRT: INPUT-RESULTS

USER INPUT

The GPRT procedure requires the following input:

. User identification line (required),

. One line for each generation or print request,

. An optional line (' +AG') which takes into account the on-line requests already entered.

Any other type of transaction is ignored.

For more details on the structure of generation-print commands, refer to the above sections.

RESULTS

There are two types of results:

. A report listing the requests,
. All printing requested.

Requests are sorted by user/library and are preceded by a 'banner' (title page).

The GPRT procedure sends a general return code:

R.C.	MEANING
4	OK with generation of source code
6	OK with generation of source code and personalized documentation or error messages
8	OK with generation of personalized documentation or error messages
10	OK without generation
12	Input-Output error
16	Sort error

NOTE: This procedure does not increment the session number.

2.3.5. GPRT: DESCRIPTION OF STEPS

GPRT : DESCRIPTION OF STEPS

GENERATION AND PRINTING: PACB

The general characteristics of this step are described in the preceding subchapter.

The generated documentation depends on the generation-printing requests taken into account. Therefore, the volume of the generated documentation and of the temporaty files is extremely variable. Banners at the beginning and at the end of user documentation, which display the user code, facilitate the distribution of printouts back to their authors.

2.3.6. GPRT: PROCESSING OF JOB STREAMS

PROCESSING THE PRINTOUTS IN RTF FORMAT (FILES xxxxx.G6)

The files generated in RTF format may require to be converted into the ISO8859 character set and to the DOS format (for UNIX platforms) before being processed by the VA Pac WorkStation.

Conversion into ISO8859 character set:

cgitrans "file   .G6" "file for PDM/RTF conversion" "character set of the platform" "iso8859"

The character set of the platform can be the following one:

"hp" on HP9000 platform

"pc850" on AIX or OS2 platform using the PC850 set

Conversion into DOS file format:

cgiux2dos "UNIX formatted file" "DOS formatted file"

These commands can be included in the PACAGP procedure.

2.3.7. GPRT: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GPRT BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        GPRT 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 "Assignment of the LG file         : /dev/null"
echo "Assignment of the LK file         : /dev/null"
echo "---------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# ***************************************************************
# * VA Pac : PRINTING AND GENERATION
# ***************************************************************
. $PACDIR/assign/$1/PAC7AB.ini
. $PACDIR/assign/$1/PAC7AC.ini
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AG.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AP.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7GS.ini
. $PACDIR/assign/$1/SQUEL.ini
PAC7ME=$PACINPUT'MBGPRT'
export PAC7ME
PAC7BM=$PACTMP'WBM'
export PAC7BM
PAC7DB=$PACTMP'GPRT.DB'
export PAC7DB
PAC7EB=$PACTMP'WEB'
export PAC7EB
PAC7EE=$PACTMP'WEE'
export PAC7EE
PAC7EG=$PACTMP'WEG'
export PAC7EG
PAC7EI=$PACTMP'WEI'
export PAC7EI
PAC7EN=$PACTMP'WEN'
export PAC7EN
PAC7EP=$PACTMP'WEP'
export PAC7EP
PAC7EQ=$PACTMP'WEQ'
export PAC7EQ
PAC7ER=$PACTMP'WER'
export PAC7ER
```

```
    PAC7EV=$PACTMP'WEV'
    export PAC7EV
    PAC7GB=$PACTMP'GPRT.GB'
    export PAC7GB
    PAC7GD=$PACTMP'GPRT.GD'
    export PAC7GD
    PAC7GE=$PACTMP'GPRT.GE'
    export PAC7GE
    PAC7GG=$PACTMP'GPRT.GG'
    export PAC7GG
    PAC7GI=$PACTMP'XGI'
    export PAC7GI
    PAC7GK=$PACTMP'ERR.GK'
    export PAC7GK
    PAC7GL=$PACTMP'ERR.GL'
    export PAC7GL
    PAC7GM=$PACTMP'XGM'
    export PAC7GM
    PAC7GN=$PACTMP'XGN'
    export PAC7GN
    PAC7G6=$PACTMP'GPRT.G6'
    export PAC7G6
    PAC7GO=$PACTMP'WGO'
    export PAC7GO
    PAC7GP=$PACTMP'GPRT.GP'
    export PAC7GP
    PAC7GQ=$PACTMP'GPRT.GQ'
    export PAC7GQ
    PAC7GR=$PACTMP'GPRT.GR'
    export PAC7GR
    PAC7GT=$PACTMP'PAW.GT'
    export PAC7GT
    PAC7GV=$PACTMP'GPRT.GV'
    export PAC7GV
    PAC7IA=$PACTMP'GPRT.IA'
    export PAC7IA
    PAC7ID=$PACTMP'GPRT.ID'
    export PAC7ID
    PAC7IH=$PACTMP'GPRT.IH'
    export PAC7IH
    PAC7II=$PACTMP'GPRT.II'
    export PAC7II
    PAC7IK=$PACTMP'GPRT.IK'
    export PAC7IK
    PAC7IL=$PACTMP'GPRT.IL'
    export PAC7IL
    PAC7IM=$PACTMP'GPRT.IM'
    export PAC7IM
    PAC7IN=$PACTMP'GPRT.IN'
    export PAC7IN
    PAC7JG=$PACTMP'WJG'
    export PAC7JG
    PAC7KB=$PACTMP'WKB'
    export PAC7KB
    PAC7KD=$PACTMP'WKD'
    export PAC7KD
    PAC7KE=$PACTMP'WKE'
    export PAC7KE
    PAC7KF=$PACTMP'WKF'
    export PAC7KF
    PAC7KG=$PACTMP'WKG'
    export PAC7KG
    PAC7KM=$PACTMP'WKM'
    export PAC7KM
    PAC7KN=$PACTMP'WKN'
    export PAC7KN
    PAC7KP=$PACTMP'WKP'
    export PAC7KP
    PAC7KQ=$PACTMP'WKQ'
    export PAC7KQ
    PAC7KR=$PACTMP'WKR'
    export PAC7KR
    PAC7KS=$PACTMP'WKS'
    export PAC7KS
    PAC7KU=$PACTMP'WKU'
    export PAC7KU
    PAC7KV=$PACTMP'WKV'
    export PAC7KV
    PAC7LG=/dev/null
    export PAC7LG
    PAC7LI=$PACTMP'WLI'
    export PAC7LI
    PAC7LK=/dev/null
    export PAC7LK
    PAC7MG=$PACTMP'WMG'
    export PAC7MG
    PAC7SO=$PACTMP'WSO'
    export PAC7SO
    PAC7WA=$PACTMP'WA'
    export PAC7WA
    PAC7W1=$PACTMP'W1'
    export PAC7W1
    PAC7W2=$PACTMP'W2'
    export PAC7W2
    PAC7W3=$PACTMP'W3'
    export PAC7W3
    PAC7W4=$PACTMP'W4'
    export PAC7W4
    PAC7W6=$PACTMP'W6'
    export PAC7W6
    PAC7W7=$PACTMP'W7'
    export PAC7W7
    PAC7W8=$PACTMP'W8'
    export PAC7W8
    PAC7W9=$PACTMP'W9'
    export PAC7W9
    SYSPAF=$PACTMP'WPAF'
    export SYSPAF
    echo "Execution : PACB"
    rtscgi PACB
    if [ -r $PACTMP'ERR.GL' ]
    then
       cp $PACTMP'ERR.GL' $PACTMP'ERR.LG'
    fi
    if [ -r $PACTMP'ERR.GK' ]
    then
       cp $PACTMP'ERR.GK' $PACTMP'ERR.LK'
    fi
    echo "End of procedure"
    echo ""
    echo "Deletion of the temporary files"
    rm -f $PACTMP'W'*
    if [ -r $PACTMP'XGI' ]
    then
       rm -f $PACTMP'XGI'
    fi
    if [ -r $PACTMP'XGM' ]
    then
```

```
      rm -f $PACTMP'XGM'
   fi
   if [ -r $PACTMP'XGN' ]
   then
      rm -f $PACTMP'XGN'
   fi
   exit 0
```

2.3.8. INTERFACE WITH GDT-PC

INTERFACE WITH GDT-PC

To enable GDT-PC to process the generation-output source files, control cards must be inserted in front of programs, maps, and copy clauses, in the following format:

```
******* <$$$> pgm_name  (map_name or copy_name resp.)
******* PACBASEPGM      (MAP or CPY resp.)
```

Example of control cards in front of program:

```
 In the PARM  user parameter management transaction
 on the screen selected by the PC DP choice ('D'
 defines the control card in front and 'P' the line
 code).

A      TITLE     TYPE : D     OPTION : P
       CONTROL CARDS IN FRONT OF PROGRAM

A  NL DESCRIPTION OF THE CONTROL CARD     S PARM.R
    1 ******* <$$$> -                       P    -
    2 ******* PACBASEPGM
```

These control cards in front (code 'P' in the example) must then be called on the entities to be generated for GDT-PC.

2.3.8.1. INTERFACE WITH WORKBENCH-MICROFOCUS

INTERFACE WITH WORKBENCH MICROFOCUS

The purpose of this interface is to split into distinct files the sources of the programs, screens or 'COPY' clauses generated by the batch server, then to write these files in a directory specified by the user.

The PACSPLIT program performs this processing.

The implementation of this option can only be done by the batch server, and by activating the command file PACAGP.

This option also requires the definition of 'BEFORE' CARDS for the VisualAge Pacbase entities to be processed.

DEFINITION OF 'BEFORE' CARDS

In order to allow the PACSPLIT program to split the source files at the generation output, it is necessary to insert 'before' cards which contains the following elements:

- Character strings specific to these lines
- Name of file to produce
- File extension
- Directory where the file will be copied

The first BEFORE CARD must contain:

```
*+++++*  Delimiter for pacsplit, between column 1
         and 7 only
filename Filename to produce
ext      Extension, on max. 3 characters
```

This information must be separated by a blank. For example:

```
*+++++* MYPROG CBL
```

The second BEFORE CARD is optional and contains the directory under which the produced files will be written. The selected directory must exist and must be accessible to the batch server.

If there is no card, the files are written under the usual directory of the batch server (with the generation output files).

The second BEFORE CARD, if it exists, contains:

```
*&&&&&*  Delimitor for pacsplit, between column
         1 and 7 only
U:+path  Directory of file writing
```

The information is separated by a blank. For example:

```
*&&&&&*  MONPROG cbl
```

That is, for the CARDS BEFORE program, for example:

The user generates from the frozen sessions and wishes to recover his generated programs under the form 'external_name'.CBL, in the (network) directory S:+COBOL+'session'+'user_code'.
Under Paclink Administrator, from the Menu screen, enter the choice PC DW and define the control cards ('D' defines the CARD BEFORE and 'W' the code of the card):

```
A   TITLE       TYPE : D    OPTION : W
    CARDS BEFORE PROGRAM WB MICRO FOCUS

A NL CONTROL CARD DESCRIPTION          S PARM.R
   1 *+++++* - cbl                       P    -
   2 *&&&&&* $PACDIR/cobol/-/-           GU   -
```

These BEFORE CARDS must then be called ('W' code in the example) in the entities to be generated.

IMPLEMENTATION IN THE BATCH SERVER

In the start-up file of the batch server, the PLBTAGP variable must be conditioned to the value YES. This allows the execution of the PACAGP command file after the generations/prints.

The PACSPLIT program must then be called in the PACAGP command file, by indicating the number of the job and the generation directory of the user.

PACAGP must therefore contain the line:

```
pacsplit %2 %3
```

PROCESSING AND ERROR MESSAGES

The PACSPLIT program processes all the files with a Gx suffix in output of the batch server (nnnnn.GP, GE, GG, ...) of %2 job, in %3 directory.

An execution report is edited in the user directory (%3) and is called 'job_number'.LOG (%2.LOG).

2.3.9. EMLD: LOADING OF USER-DEFINED ERROR MESSAGES

2.3.9.1. EMLD: INTRODUCTION

EMLD: LOADING OF USER-DEFINED ERROR MESSAGES

EMLD: INTRODUCTION

The EMLD procedure performs the initial loading of user- defined error messages. These messages are obtained from the sequential output file of the GPRT procedure (file with the GL suffix).

EXECUTION CONDITION

The GPRT procedure must first be run with an error message generation request.

Batch procedure authorization option: Required authorization level is 2.

USER INPUT

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

2.3.10. EMLD: DESCRIPTION OF STEPS

EMLD: DESCRIPTION OF STEPS

```
INDEXED LOADING OF USER-DEFINED ERROR MESSAGES: PACL93

.Input files:
 -Input transactions
  PAC7MB
 -Sequential user-defined error messages
  PAC7GL
 -Data file
  PAC7AR
 -VisualAge Pacbase error messages
  PAC7AE

.Permanent output file:
 -User-defined error messages, indexed
  PAC7EM

.Output reports:
 -Execution report
  PAC7IY
 -Batch procedurest geemup denon option
  PAC7DD

.Return code:
 8:  Unauthorized user
```

2.3.11. EMLD: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) EMLD BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        EMLD 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 $PACDIR/batch/proc/MSGPAUSE.ini
# *********************************************************
# * VA Pac : LOADING OF USER-DEFINED ERROR MESSAGES
# *********************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# *********************************************************
PAC7LG=$PACTMP'ERR.GL'
export PAC7LG
if [ ! -f "$PAC7LG" ]
then
 echo "The $PAC7LG file does not exist,"
 echo "the generated error message file must be copied into the"
 echo "$PAC7LG file"
 exit 1
fi
PAC7GL=$PACTMP'ASCII.GL'
export PAC7GL
echo "Execution : PTUSGL"
rtscgi PTUSGL
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7AR.ini
 PAC7MB=$PACINPUT'MBEMLD'
 export PAC7MB
 PAC7EM=$PACTMP'ERRMSG'
 export PAC7EM
 PAC7GL=$PACTMP'ASCII.GL'
 export PAC7GL
 PAC7IY=$PACTMP'EMLDIY.L93'
 export PAC7IY
 PAC7DD=$PACTMP'EMLDDD.L93'
 export PAC7DD
 echo "Execution : PACL93"
```

```
     rtscgi PACL93
     RETURN=$?
     case $RETURN in
     0)
      echo "End of procedure"
      ;;
     8)
      echo "Error in executing PACL93"
      echo "Error 8 : Error on * input line"
      ;;
     *)
      echo "Error in executing PACL93"
      ;;
     esac
     ;;
    *)
     echo "Error in executing PTUSGL"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

2.3.12. EMUP: UPDATE OF USER-DEFINED ERROR MESSAGES

2.3.12.1. EMUP: INTRODUCTION

EMUP: UPDATE OF USER-DEFINED ERROR MESSAGES

EMUP: INTRODUCTION

The EMUP procedure updates the User-Defined Error Message file. These messages are obtained from the sequential output file of the GPRT procedure (file with a GL suffix) or from transactions for error message deletions at the entity level.

EXECUTION CONDITION

The User-Defined Error Message file must exist.

Before creating or modifying error messages, the GPRT procedure must be executed with a request to generate error messages.

Batch procedure access authorization option: Level 2 is required.

2.3.13. EMUP: USER INPUT

EMUP : USER INPUT

A line '*' per library containing entities which message(s) must be deleted:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Library code

One command line per entity for which error message deletion is requested:

POS.	LEN.	VALUE	MEANING
1	1	'D'	Transaction code (deletion)
2	2		Entity type; same as in CHOICE field
		'O '	Screen
		'D '	Data structure
		'S '	Segment
4	6		Entity code

2.3.14. EMUP: DESCRIPTION OF STEPS

EMUP: DESCRIPTION OF STEPS

```
SORT ON GENERATED SEQUENTIAL ERROR MESSAGES:  PTUSGL

.Input file:
 -Sequential user error messages:  PAC7LG
  (ERR.GL file in the temporary directory,
   output from GPRT)

.Ouptu file:
 -Sorted sequential error messages:  PAC7GL
  (ASCII.GL file in temporary directory)


UPDATE OF USER-DEFINED ERROR MESSAGES: PACL92

.Input files:
 -Sequential user-defined error messages
  (ASCII.GL in the temporary directory)
  PAC7GL
 -Data file
  PAC7AR
 -VisualAge Pacbase error messages
  PAC7AE
 -Transaction file
  PAC7MB

.Permanent output file:
 -User-defined error message indexed file
  PAC7EM

.Output reports:
 -Transaction report
  PAC7IU
 -Error message report
  PAC7IX
 -Batch-procedure authorization option
  PAC7DD

.Return code:
 8:  Unauthorized user
```

2.3.15. EMUP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) EMUP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        EMUP 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 $PACDIR/batch/proc/MSGPAUSE.ini
# **************************************************************
# * VA Pac : UPDATE OF USER-DEFINED ERROR MESSAGES FILE
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE '*' LINE WITH USER CODE, PASSWORD AND CODE OF THE
# *  LIBRARY CONTAINING MESSAGES TO BE DELETED
# * .ONE LINE PER ENTITY (DELETION OF ITS ERROR MESSAGES)
# *  COL  1    : 'D' TRANSACTION CODE (DELETION)
# *  COL  2-3  : ENTITY TYPE
# *            : 'O ' SCREEN
# *            : 'D ' DATA STRUCTURE
# *            : 'S ' SEGMENT
# *  COL  4-9  : ENTITY CODE
# **************************************************************
PAC7LG=$PACTMP'ERR.GL'
export PAC7LG
if [ ! -f "$PAC7LG" ]
then
 echo "The $PAC7LG file does not exist,"
 echo "the generated error message file must be copied into the"
 echo "$PAC7LG file"
 exit 1
fi
PAC7GL=$PACTMP'ASCII.GL'
export PAC7GL
echo "Execution : PTUSGL"
rtscgi PTUSGL
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7AR.ini
 PAC7MB=$PACINPUT'MBEMUP'
 export PAC7MB
 PAC7EM=$PACTMP'ERRMSG'
 export PAC7EM
```

```
     PAC7GL=$PACTMP'ASCII.GL'
     export PAC7GL
     PAC7IU=$PACTMP'EMUPIU.L92'
     export PAC7IU
     PAC7IX=$PACTMP'EMUPIX.L92'
     export PAC7IX
     PAC7DD=$PACTMP'EMUPDD.L92'
     export PAC7DD
     echo "Execution : PACL92"
     rtscgi PACL92
     RETURN=$?
     case $RETURN in
     0)
      echo "End of procedure"
      ;;
     8)
      echo "Error in executing PACL92"
      echo "Error 8 : Error on * input line"
      ;;
     *)
      echo "Error in executing PACL92"
      ;;
     esac
     ;;
    *)
     echo "Error in executing PTUSGL"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

2.3.16. PPAF: PAF PRE-PROCESSOR

2.3.16.1. PPAF: INTRODUCTION

PPAF: PAF PRE-PROCESSOR OF GENERATED PROGRAMS

PPAF: INTRODUCTION

Using PAF operators, the PPAF procedure processes generated user programs containing SQL requests for access to the Database.

EXECUTION CONDITIONS

None.

IMPLEMENTATION

This procedure may be executed in different ways:

- Either after program generation using the GPRT procedure, whose output is retrieved and used as input to PPAF, before compilation or storage in a source program library,

- Or by requesting the procedure in the Control Cards in front/in back of generated program; the appropriate JCL must have been previously entered in the selected options (PC screen). The input consists in updating the user parameters in TP mode or via the PARM batch procedure.

2.3.17. PPAF: USER INPUT

PPAF: USER INPUT

USER INPUT

User input is the COBOL source code of programs containing PAF operators to be processed by the pre-processor before compilation.

After the IDENTIFICATION DIVISION, each program contains a command line for the pre-processor. Its structure is as follows :

POS.	LEN.	VALUE	MEANING
1	6	nnnnnn	COBOL line number
7	1	'*'	Comment
8	5	'TP '	On-line program OR
		'BATCH'	Batch program
13	6	'LIB:'	Fixed label
19	3	bbb	Library code
22	1	blank	Not used
23	5	nnnns	Session number - Session version
28	1	blank	Not used
29	2	--	Generation variant(s)
31	5	'AR:'	Fixed label
36	1	1	Database language code
37	5	'SC:'	Batch Language program skeleton
		'SG:'	On-line program skeleton
		'SR:'	COBOL program skeleton
42	1	1	Skeleton language
43	1	blank '	Not used
44	6	'SINGLE'	Single quotes OR
		'DOUBLE'	Double quotes

EXAMPLES

```
000020*TP    LIB: APP 2345   00 AR: F SG: F SINGLE

000020*BATCH LIB: APP 2300T  4  AR: F SC: F DOUBLE
```

This line is automatically generated by the GPRT procedure.

PRINTED OUTPUT

This procedure prints an error report.

RESULT

The result of the PPAF procedure is the COBOL source in which PAF operators have been processed and calls to PAF batch or on-line sub-programs have been generated.

2.3.18. PPAF: DESCRIPTION OF STEPS

PPAF: DESCRIPTION OF STEPS

```
PREPROCESSOR: PAFP10

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error message file
  PAC7AE

.Input file:
 -Generated programs
  PAF80

.Output files:
 -Generated programs to be compiled
  COB80

.Output report:
 -Execution report
  PAFREP
```

2.3.19. PPAF: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) PPAF BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "----------------------------------------------------------"
echo "                        PPAF PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
echo "  (PAF = input COBOL; COB = pre-compilation COBOL)"
. $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 : VA Pac ACCESS FACILITY PRE-PROCESSING
# ***************************************************************
# * INPUT TRANSACTION FORMAT :
# * .COMMAND LINE FOR PREPROCESSOR
# *  COL  1-5  : COBOL LINE NUMBER
# *  COL  7    : '*'  COMMENT
# *  COL  8-12 : 'TP   ' OR 'BATCH'
# *  COL 14-17 : 'LIB:'
# *  COL 19-21 : LIBRARY CODE
# *  COL 23-27 : SESSION NUMBER AND VERSION
# *  COL 29-30 : GENERATION VARIANT
# *  COL 32-34 : 'AR:'
# *  COL 36    : DATABASE LANGUAGE CODE
# *  COL 38-40 : 'SC:' BATCH PROGRAM SKELETON
# *            : 'SG:' ON-LINE PROGRAM SKELETON
# *            : 'SR:' COBOL PROGRAM SKELETON
# *  COL 42    : SKELETON LANGUAGE CODE
# *  COL 44-49 : 'SINGLE' OR 'DOUBLE' (QUOTES)
# ***************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAF80=$PACTMP'PAF'
export PAF80
COB80=$PACTMP'COB'
export COB80
PAFREP=$PACTMP'PAFREP'
export PAFREP
echo "Execution : PAFP10"
rtscgi PAFP10
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
```

```
     ;;
    *)
     echo "Error in executing PAFP10"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

2.4. PACX: EXTRACTION FROM THE VA PAC DATABASE

2.4.1. PACX: INTRODUCTION

PACX: INTRODUCTION

The PACX procedure extracts data from the VisualAge Pacbase Database in the form of transactions. These transactions can then be used as input to one of the following procedures:

```
    . UPDT
    . UPDP
    . CPSN (If the optional LCU PARTITIONED DATABASE
           MANAGER utility is available.)
```

EXECUTION CONDITIONS

None, since the database is not directly updated by this procedure.

The authorization level is specified for each extractor.

2.4.2. PACX: USER INPUT COMMON TO ALL EXTRACTORS

PACX: USER INPUT COMMON TO ALL EXTRACTORS

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Extraction-library code, or target-library code if RMEN with upload
22	4	nnnn	Session number (blank=current ses.)
26	1	T	Session status if Test session
28	1	l	Language code (A=english, F=french)
29	4	cccc	Extractor code
33	1	'1'	Formatting for UPDT
		' '	No formatting for UPDT
34	1	'1'	Formatting for UPDP (PAF)
		' '	No formatting for UPDP (PAF)
35	1	'1'	Formatting for CPSN
		' '	No formatting for CPSN
40	3	ppp	DSMS Product Code
43	6	nnnnnn	DSMS Change number (DSMS Function only)
49	1		Lock processing
		' '	Lock extraction: user code = '*'-line user code
		'1'	No lock extraction
		'2'	Lock extraction: user code = original user code
50	1	' '	No transfer of password
		'1'	Password transfer
69	3	bbb	Library code for the '*'-line of the output file(s) (For EXTR,EXLI, and EXUE only)
76	5	nnnnT	Session number for the '*'-line of the output file(s) (For EXTR,EXLI, and EXUE only)

Possible values for the extractor code include:

```
  - EXLI:  Extraction of libraries or library sub-networks
  - EXTR:  Extraction of entities
  - EXTA:  Extraction of entities (extracted transactions
           are sorted, according to the input
           identification lines order. So if each request is
           preceded by a '*' line, extracted transactions
           will be sorted in the order of the requests).
           The formatting is forced to UPDT.
  - EXPJ:  Extraction of Journal (formatting for CPSN is not
           possible)
  - EXPU:  Extraction of entities to be purged (formatting
           for CPSN is not possible)
  - EXUE:  Extraction of UEO's.
  - RMEN:  Extraction of entitites for upload/replacement/
           recoding (formatting for CPSN is not possible).
           RMEN is subject to a separate purchase agreement.
```

IMPORTANT:

- One extractor type only for each run: If the procedure detects more than one type of extractors, it will take only the first one into account.

- One formatting type only for each run: If the procedure detects more than one type of formatting, it will take only the first one into account.

- Formatting for CPSN: This procedure is part of the LCU Partitioned Database Manager optional utility. Its use is therefore subject to a special licence contract.

- Maximum number of input '*' cards : 99

PRINTED RESULT:

The PACX procedure produces:

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

2.4.3. EXLI: LIBRARY EXTRACTION

2.4.3.1. EXLI: INTRODUCTION

EXLI: LIBRARY EXTRACTION

EXLI: INTRODUCTION

The EXLI procedure extracts a complete library from the database and transforms it in transactions which are used in the update or comparison procedures.

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

EXECUTION CONDITIONS

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

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

2.4.4. EXLI: USER INPUT

EXLI: USER INPUT

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

PRINTED OUTPUT

The extractor prints:

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

. The details of records extracted for each library.

IMPORTANT RECOMMENDATIONS

The sub-network can include up to 25 libraries: 25 '*'-lines maximum.

The order of extraction requests must be the same as the sub-network description 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 one request is invalid, all requests are rejected.

NOTE: When the EXLI output is to be processed by CPSN, in order to ensure complete coherence in the comparison, it is necessary to extract a complete sub-network.

EXAMPLE

```
           SUB-NETWORK          EXTRACTION TRANSACTIONS
             AAA (1)
            .       .           (1) _*USERCODEPASSWORDAAA
         .           .          (2) _*USERCODEPASSWORDXXX
         XXX (2)   MMM (7)      (3) _*USERCODEPASSWORDDDD
        .   .                   (4) _*USERCODEPASSWORDEEE
      .       .                 (5) _*USERCODEPASSWORDKKK
   DDD (3)   EEE (4)            (6) _*USERCODEPASSWORDRRR
            .   .               (7) _*USERCODEPASSWORDMMM
          .       .
       KKK (5)   RRR (6)
```

2.4.5. EXTR: ENTITY EXTRACTION

2.4.5.1. EXTR: INTRODUCTION

EXTR / EXTA : ENTITY EXTRACTION

EXTR / EXTA : INTRODUCTION

The EXTR extractor type allows for selection of the whole entities or parts of entities.

If the request is of the 'ALL' type, the whole entity is extracted, i.e. the entity itself but also all the entities it uses, as well as entities used by those, and so on. Used entities that are not cross-referenced are not extracted.

Depending on the type of formatting requested, the resulting file can be used as input to the UPDT, UPDP or CPSN procedures (if the request is of the 'ALL', 'ONLY' or 'EXPT' type; the formatting for CPSN is not allowed). For EXTA, the formatting is forced to UPDT.

It is therefore possible to compare entities.

EXECUTION CONDITIONS

None, since the database is not directly updated.

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

2.4.6. EXTR: USER INPUT

EXTR / EXTA : USER INPUT

```
USER INPUT
One or two command lines per entity to be extracted.

First line :
------------------------------------------------------------
!Pos.! Len.! Value  ! Meaning                               !
!----+-----+--------+---------------------------------------!
!  2 !   1 ! 'W'    ! Line code                             !
!  3 !   1 ! '1'    ! Line number                           !
!  4 !   2 ! 'EX'   !                                       !
!  6 !   1 !        ! Library selection code:               !
!    !     ! 'U'    ! Library alone                         !
!    !     ! 'C'    ! Library and its upper-level libraries!
!    !     ! '+'    ! Library and its upper-level libraries!
!    !     !        ! with identification lines ('*' lines)!
!    !     !        ! generation                            !
!  7 !  25 ! Choice ! Entity to be extracted, coded in the !
!    !     !        ! same way as the 'Choice' field in TP.!
! 32 !   4 !        ! Extraction type:                      !
!    !     ! '    ' ! Entity alone (required for EXTA)     !
!    !     ! 'ALL ' ! Entity and used entities              !
!    !     ! 'ONLY' ! Entity and only used entities whose  !
!    !     !        ! types are specified in the following !
!    !     !        ! part of the line                      !
!    !     ! 'EXPT' ! Entity and used entities, except     !
!    !     !        ! those whose types are specified in   !
!    !     !        ! the following part of the line       !
! 36 !     !        ! 15-position table (3 char./position) !
!    !     !        ! containing exceptions or selections  !
!    !     !        !    'DEL':  Data Element               !
!    !     !        !    'DBD':  Database Block             !
!    !     !        !    'DST':  Data Structure             !
!    !     !        !    'SEG':  Segment                    !
!    !     !        !    'RPT':  Report                     !
!    !     !        !    'TXT':  Text                       !
!    !     !        !    'VOL':  PDM Volume                 !
!    !     !        !    'MAN':  User Manual                !
!    !     !        !    'PGM':  Program                    !
!    !     !        !    'DLG':  Dialog                     !
!    !     !        !    'SCR':  Screen                     !
!    !     !        !    'PIA':  P.I.A.                     !
!    !     !        !    'MET':  Methodology                !
------------------------------------------------------------
```

First line (continued)

			'UEN': User Entity
			'URE': User-defined Relationship
			'$tt': User Entity Occurrence (tt = occurr. type code)

Second line (continuation line for selections and exceptions):

Pos.	Len.	Value	Meaning
2	1	'W'	Line code
3	1	'2'	Line number
36			15-position table (3 characters per position) containing the exceptions or selections

(*) The EXTR procedure also works with choices that are specific to the WorkStation. These choices must be entered from the eigth position, in the following way:

_W1EX_U//A_CCCXXXXXX

where A is the methodology code and CCC the entity local code.

The use of the 'multi-layered extractor' option ('ALL', 'EXPT' or 'ONLY' extraction type) is subject to a purchase agreement. For EXTA, this field value must be blank.

If the extraction type is not specified, the extraction of a Data Structure extracts the Data Structure only. This field must therefore be completed if Segments (or Reports) for that Data Element are to be extracted also. Similarly, for a Dialog and its Screens, or a User Entity and its Occurrences, this field must be completed.

The extraction stops at the first selection or exclusion level.

Example: Extraction of a Program with 'EXTPSEG' - The Data Elements used by Segments used by the Program are not extracted since the extractor does not consider those segments.

PRINTED OUTPUT

The procedure produces:

. A list of extracted entities:
- Sorted for EXTR,
- In the order of the requests for EXTA.

2.4.7. EXPJ: TRANSACTION EXTRACTION FROM THE JOURNAL

2.4.7.1. EXPJ: INTRODUCTION

EXPJ: INTRODUCTION

The EXPJ procedure has a two-fold action:

. It converts the Journal file into update transactions with possible selection from a range of dates, sessions, libraries, etc.

. It prints out a listing of the contents of the archived Journal file, using the same criteria.

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

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

EXECUTION CONDITIONS

Batch procedure access authorization option:
. level 2 is required.

Password transfer option ('*'-line col. 50 = 1):
. database access authorization level 4 is required.

2.4.8. EXPJ: USER INPUT

EXPJ: USER INPUT

USER INPUT

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

POS.	LEN.	VALUE	MEANING
2	1	'J'	Line code
3	1	'S'	Selection on session number
		'D'	Selection on date
4	1	' '	Chronological sort
		'N'	No chronological sort
5	1	' '	Sort by user
		'N'	No sort by user
6	1	' '	Sort by Library
		'N'	No sort by library
7	8	uuuuuuuu	User code for batch update
15	8	pppppppp	User password
23	4	dddd	Session number: beginning (if 'S')
27	4	ffff	Session number: end (if 'S')
31	8	CCYYMMDD	Date of beginning of select.(if 'D')
39	8	CCYYMMDD	Date of end of selection (if 'D')
47	1		Version of selected transactions
		' '	Selection of all sessions
		'Z'	Selection of current session
		'T'	Selection of frozen session
48	3	'bbb'	Code of selected library
51	5	'ssssT'	Selection of T-type session (test version of frozen session:'ssssT')
56	3	ppp	DSMS Product Code
59	6	nnnnnn	DSMS Change number (Selection by change number-DSMS)
65	6	HHMMSS	Starting time
71	6	HHMMSS	Ending time

REPORTS

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

RESULT

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

2.4.9. EXPU: EXTRACTION OF UNUSED ENTITIES FOR PURGE

2.4.9.1. EXPU: INTRODUCTION

EXPU: INTRODUCTION

The EXPU utility purges unused entities from a database.

Two types of purges are possible:

-'Logical' purge of entities which have become obsolete;

-'Physical' purge of entities which have never been used.

TERMINOLOGY

FINAL ENTITIES:

These entities, which are not used by other entities, include:

. Programs ('P' entity);

. Screens, C/S screens, Business components,..('O' entity) ('O' entity);

. User manuals ('U' entity);

. Volumes ('V' entity);

. User Entity Occurrences ('$' entity);

. Database blocks ('B' entity).

FREE-TYPE CROSS-REFERENCE:

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

PRINCIPLES

LOGICAL PURGE:

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

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

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

PHYSICAL PURGE:

The EXPU procedure informs the user of the entities which have never had any cross-references since their creation in a given context. For these entities, physical purge transactions are generated. These transactions can be used as input to the REOR procedure.

NOTE: THE LIBRARY ENTITY IS NOT PROCESSED.

EXECUTION CONDITIONS

Batch procedure access authorization option:
. Authorization level 3 is required.

2.4.10. EXPU: USER INPUT

EXPU: USER INPUT

USER INPUT

One line with the extraction characteristics:

POS.	LEN.	VALUE	MEANING
2	2	'P '	Line code
4	1		Type of purge:
		'P'	Physical (via the REOR procedure)
		'L'	Logical (via the UPDT procedure)
5	1		Search option for the entity definition screens:
		'U'	In the indicated library only
		'Z'	In the indicated library and corresponding sub-network
6	4	ssss	Session number (type 'L' only) from which the entities must not be used in order to be purged
10	3	ttt	Entity type
13	6	pppppp	Program code (program processing only)
19	1	1	Allows the removal of purge transactions which are not cross-referenced in the sub-network nor in the next higher network.

COMMENTS

Each 'ENTITY TYPE' may be processed separately. If the 'EN- TITY TYPE' field is not entered, all entities are processed EXCEPT the FINAL ENTITIES.

Command Examples:

```
        *user    passwordBIB
        P PZ    E
```

Command for physical purge transactions for the data elements in the BIB library sub-network.

```
        *user    passwordBIB
        P LU2222P  PROGR
```

Command for logical deletion transactions for the programs in the BIB library whose codes are less than or equal to PROGR, starting from session number 2222.

```
        *user    passwordBIB
        P PU
```

Command for physical purge transactions for all entities in the BIB library (except the FINAL ENTITIES).

PRINTED OUTPUT

This procedure prints out:

- A list of the entities to be purged logically,

- A list of the entities to be purged physically.

RESULT

The result of this procedure is:

- In the case of a logical purge, a sequential file containing entity deletion transactions to be used as input in the Database updating (UPDT) procedure.

These transactions are sorted as follows:

. By decreasing hierarchical library level

. By library

. By record type: descriptions, definition screens.

- In the case of a physical purge, a sequential file containing entity purge transactions to be used as input to the Reorganization (REOR) procedure.

Each transaction contains a maximum of six entities to be purged.

For each entity, the following information is included:

. The entity type

. The entity code

. The library code. (See Chapter "REOR: Database Reorganization", Subchapter 'INPUT-RECOMMENDATIONS', in the Administrator's Guide.)

2.4.11. EXUE: EXTRACTION OF USER ENTITIES

2.4.11.1. EXUE: INTRODUCTION

EXUE: INTRODUCTION

The EXUE procedure extracts user entity occurrences according to their type code, formatted as simple records in a sequential file.

The EXUE procedure is part of the Dictionary Extensibility Function which is an optional component and whose use depends upon the corresponding purchase agreement.

EXECUTION CONDITIONS

Batch-procedure access authorization option:
. Level 2 is required.

2.4.12. EXUE: USER INPUT

EXUE: USER INPUT

USER INPUT

One command line per user entity:

POS.	LEN.	VALUE	MEANING
2	4	W1EX	Line code
6	1	$	UEO Extraction identifier
7	1		Library selection code:
		U	Selected library
		C	Selected library + higher level libr.
8	2	CC	User Entity type code

REPORT

The EXUE procedure prints a list of the extracted UEOs.

RESULT

The output of the EXUE procedure is a sequential file with a fixed format in which the contents of the selected user entity occurrences are recorded.

The length of each record is 112 characters.

Each record includes:

. A common part containing all the characteristics necessary to identify each extracted line.

. A specific part whose format depends on the user entity description.

2.4.13. RMEN: RENAME/MOVE OF ENTITIES

2.4.13.1. RMEN: INTRODUCTION

RMEN: ENTITY RENAMING / MOVING

RMEN: INTRODUCTION

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

Through the RMEN procedure you can:

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

This procedure may be applied to Dictionary and entities.

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

EXECUTION CONDITIONS

None, since the Database is not directly updated.

Batch procedure access authorization option:
. Level 3 is required.

Only users with a authorization level 4 on the database can use this procedure.

To rename (RN) or replace (RP) entities, an authorization level 4 on the library in which the entity is found is sufficient.

2.4.14. RMEN: USER INPUT

RMEN: USER INPUT

Batch procedure access authorization:

One or more command lines per entity to be processed:

First line

POS.	LEN.	VALUE	MEANING
2	2	W2	Line code
4	2		Processing option:
		MV	Entity move (UP)
		RN	Entity rename
		RP	Entity replace
		MR	Upward move and rename
6	3	ttt	Entity type or local code of a WorkStation entity: D, E, I, O, P, R, S, T, $nn, F, M, Q, B, V, or SDO, RUB ...
9	6	elemt1	Code of entity to be extracted
15	1		Separator blank
16	3	sss	Source library code (for MOVE)
19	1		Separator blank
20	6	elemt2	Entity code after RENAME, or code of replacing entity in case of REPLACE
26	6	elemtP	Parent Data Element code
32	3	'ALL'	for 'MV' and 'MR': Selects all occurrences of a UE or all Segments or Reports of a Data Structure (implicit option for 'RN' and 'RP')
35	3		For extraction of WorkStation entities: methodology code
		'//A'	SSADM
		'//M'	MERISE
		'//D'	YSM
		'//O'	OMT
		'//F'	IFW

First line (continued):

POS.	LEN.	VALUE	MEANING
38	3		REPLACE: Selection of the types of the entities to be modified 'DEL': Data Element 'DBD': Database Block 'DST': Data Structure 'SEG': Segment 'RPT': Report 'TXT': Text 'VOL': PDM volume 'MAN': User Manual 'PGM': Program 'SCR': Screen 'PIA': P.I.A. 'MET': Methodology 'UEN': User Entity 'URE': User-defined Relationship '$tt': User Entity Occurrence : (tt = occurrence type code) '$**': All UEOs
41	6		REPLACE: Codes of entities to be modified (* may be used if you want to specify only the beginning of a code.

Lines for REPLACE (continuation lines for selection):

POS.	LEN.	VALUE	MEANING
2	2	'W2'	Line code
4	2	'RP'	'REPLACE'
6	3	'*'	'continuation line'
38	3		Selection of types of entities to be modified
41	6		Codes of entities to be modified

REQUEST-SEQUENCING REQUIREMENTS

The sequencing of RMEN requests should follow a logical order, e.g.:

A parent Data Element must be moved to the higher-level library BEFORE its child Data Element(s).

When a Segment is called by another Segment, the called Segment must be moved to the higher-level library BEFORE the Segment that is calling it.

When a macro-structure is called by a batch Program or on-line Screen, it must be moved into the higher-level library BEFORE this Program or Screen.

REQUEST-INPUT REQUIREMENTS

All input is required except:

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

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

EXECUTION RULES

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

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

PRINTED OUTPUT

This procedure prints out the following:

. The list of entities processed by RMEN.

. The number of lines extracted for each request.

RESULT

The output is a sequential file which contains update transactions:

. Creation or modification transactions sorted by:

- Ascending library hierarchical level,
- Library,
- Record type (uses, definition, or description).

. Deletion transactions sorted by:

- Descending library hierarchical level,
- Library,
- Record type (uses, description, definition).

NOTES:

The replacement of entities (RP) does not ensure data consistency. Thus, if you replace a Data Element with another one in a Segment, RMEN does not modify the program lines where this Data Element is used by this Segment, except if you have requested the replacement in programs.

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

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

2.4.15. RMEN: RECOMMENDATIONS AND RESTRICTIONS

RECOMMENDATIONS AND RESTRICTIONS

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

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

ALL ENTITY TYPES

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

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

. When an occurrence's General Documentation contains PIA or User Relation calls, its cross-referenced occurrences must be in a library whose level is greater or equal to that of the target library.

. When an occurrence is renamed, if it is called on Assigned Text (-AT) lines, it is changed on I-type lines, but not on J-type lines.

DATA STRUCTURES

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

CAUTION :

An upward move of a Data Structure involves the upward move of all of its Segments and Reports contained in the source library in cases where the GLOBAL UPWARD MOVE field contains 'ALL'. If this field is blank, the Segments and Reports remain in the source library.

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

SEGMENTS AND REPORTS

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

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

For Reports, the existence validation is performed for called Data Elements only.

DATA ELEMENTS

The indication of a parent Data Element code affects only the Data Element Definition in the source library. By default, a child Data Element remains attached to its parent. However, it is possible to suppress this link by entering the code '&&&&&&' in the parent Data Element field.
A child Data Element can be turned into a parent Data Element or may be assigned another parent by specifying a parent Data Element code. This parent Data Element must be defined in a library upper or equal to the target library.

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

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

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

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

PROGRAMS

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

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

SCREENS

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

USER ENTITIES

A User Entity can be processed only if there is no other User Entity bearing the same call code in the sub-network of the target library.

CAUTION :

When the GLOBAL UPWARD MOVE field contains 'ALL', an upward move of a User Entity involves the upward move of all of its occurrences contained in the source library. If this field is blank, the occurrences remain in the source library.

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

USER ENTITY OCCURRENCES (UEOs)

The existence of the User Entity in a library higher or equal to that of the target library is checked, as is that of occurrences linked to the UEO via User Relations.

PACMODEL ENTITIES

For PACMODEL Objects and Elements/Properties called in description screens ('-CM' and '-CE'), an existence check is performed in the library whose level is higher than or equal to that of the target library.

DATABASE BLOCKS

The existence of PACMODEL Objects or Called Segments is checked.

VOLUMES

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

OCCURRENCES MANAGED VIA THE WORKSTATION

Calls of the '//M', '//Y' and '//D' type are used to extract all the WorkStation entities. The local entity type -- 3-character code -- must be entered (in the ENTITY TYPE field) as well as the code of entity before processing, the library code and the code of the entity after processing. The WorkStation methodology (MERISE, IFW, OMT, YSM...) is entered in a special field at position 35 in the 'W2' user input line.

NOTE: One RMEN execution can process occurrences related to only one Methodology.

2.4.16. PACX: DESCRIPTION OF STEPS

PACX: DESCRIPTION OF STEPS

```
EXTRACTION:  PACX

This step extracts transactions according to user input.

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE
 -Archived transactions
  PAC7PJ

.Input transaction file:
 -User input
  PAC7MB (MBPACX file in INPUT directory)

.Work files:
 -User input
  PAC7BM
 -EXPU work file
  PAC7MM
 -EXPJ work file
  PAC7MJ
 -RMEN work file
  PAC7TE
 -RMEN work file
  PAC7RE
 -RMEN work file
  PAC7RM


 -Extracted transactions
  PAC7WD
 -Multi-layered Extractor work file
  SYSEXT

.Output files:
 -Extracted transactions for UPDT
  PAC7MV (PACX.MV in the Database TMP directory)
 -Extracted transactions for REOR (EXPU)
  PAC7MR (PACX.MR in the Database TMP directory)


 -Extracted transactions for UPDP
  PAC7GY (PACX.GY in the Database TMP directory)
 -Extracted transactions for CPSN
  PAC7TD (PACX.TD in the Database TMP directory)
 -Extracted transactions for EXUE
  PAC7UE (PACX.UE in the Database TMP directory)

.Output reports:
 -General printout of the program stream
  PAC7IA
 -List of errors on input transactions
  PAC7DD
 -Summary reports on extractions


  PAC7EE
  PAC7EP
  PAC7EQ
  PAC7EZ

.Sort file(s):
  Not assigned


  0:  No error
  8:  Serious error (detailed in PAC7DD)
```

2.4.17. PACX: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) PACX BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "----------------------------------------------------------"
echo "                        PACX 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 : EXTRACTIONS
# *********************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PACSAVPJ.ini
PAC7PJ=$PACSAVPJ
export PAC7PJ
PAC7MB=$PACINPUT'MBPACX'
export PAC7MB
PAC7BM=$PACTMP'WBM'
export PAC7BM
PAC7WD=$PACTMP'WWD'
export PAC7WD
PAC7MM=$PACTMP'WMM'
export PAC7MM
PAC7MJ=$PACTMP'WMJ'
export PAC7MJ
PAC7TE=$PACTMP'WTE'
export PAC7TE
PAC7RE=$PACTMP'WRE'
export PAC7RE
PAC7RM=$PACTMP'WRM'
export PAC7RM
PAC7UE=$PACTMP'PACX.UE'
export PAC7UE
PAC7GY=$PACTMP'PACX.GY'
export PAC7GY
PAC7TD=$PACTMP'PACX.TD'
export PAC7TD
PAC7IA=$PACTMP'PACX.IA'
export PAC7IA
PAC7DD=$PACTMP'PACX.DD'
export PAC7DD
PAC7ED=$PACTMP'PACX.ED'
```

```
export PAC7ED
PAC7EE=$PACTMP'PACX.EE'
export PAC7EE
PAC7EZ=$PACTMP'PACX.EZ'
export PAC7EZ
PAC7EP=$PACTMP'PACX.EP'
export PAC7EP
PAC7EQ=$PACTMP'PACX.EQ'
export PAC7EQ
PAC7MV=$PACTMP'PACX.MV'
export PAC7MV
PAC7MR=$PACTMP'PACX.MR'
export PAC7MR
SYSEXT=$PACTMP'WPACX.SY'
export SYSEXT
rm -f $SYSEXT*
echo "Execution : PACX"
cobrun PACX
RETURN=$?
case $RETURN in
0)
 ;;
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
 ;;
*)
 echo "Error in executing PACX"
 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'W'*
```

3. PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION

3.1. XPAF: EXTRACTION MASTER PATH

3.1.1. XPAF: INTRODUCTION

XPAF: INTRODUCTION

PRINCIPLES

The Extraction Master Path validation procedure, XPAF, allows for the simulation of specific extractions that the standard procedures are not able to perform.

RESULTS

The type of result depends on whether or not the extracted domain is to be integrated into a report : Macro-Command or User Extraction program.

Macro-Command: a subroutine to be activated during a printing request by GPRT (choice: PCV).

User Extraction program: a Source Program to be compiled and executed.

PREREQUISITE

In order to use this procedure, the system manager must update the Database with the transaction file supplied for installation which contains the .PPTEX User Entity, whose call code is 7E.

IMPLEMENTATION

Before the procedure can be executed, the user must define an occurrence of this user entity ($7E). Its definition file and description determine the characteristics and format of the general extraction program.

EXECUTION CONDITIONS

Extraction Master Path users must have at least a level 2 authorization on the Database.

ABNORMAL EXECUTION

For any type of abnormal end the procedure can be re-executed once the problem has been solved.

PRINTED OUTPUT

This procedure prints a validation report and a simulation of the Extraction Master Path.

3.1.2. XPAF: USER INPUT

XPAF: USER INPUT

One '*' line per library and session to be consulted

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	nnnn	Session number
26	1	T	Session version
68	1	' '	Standard print
		'1'	Uppercase print

One command line 'EX' for the following elements:

POS.	LEN.	VALUE	MEANING
2	2	'EX'	Line code
4	2		Call code (7E by default)
6	6	eeeeee	User Entity occurrence code
Warning:			Specify library and session if the UEs whose occurrences are to be extracted in a parallel sub-network (UEOs extractions managed by the WorkStation for example)
12	3	bbb	Library code
15	4	nnnn	Session number
19	1	T	Session version
20	6	'UPDATE'	Update of GS
		or	
		SPACE	Check of the presence of the master path in GS. Check of the user entity occurence's use in the sub-network. No update of GS if presence or use.

EXAMPLES

```
*user    passwordBIB
EX7EEXT001        UPDATE
*user    passwordBIB
EX7EEXT002
```

3.1.3. XPAF: DESCRIPTION OF STEPS

XPAF: DESCRIPTION OF STEPS

```
ACCESS AND VALIDATION:  PTEX30

.Input files:
 -VA Pac error-message file
  PAC7AE
 -Index file
  PAC7AN
 -Data file
  PAC7AR

.Input transaction file:
 -User input
  PAC7MB

.Permanent input file:
 -Variable skeleton-file
  PAC7SP

.Permanent input/output file:
 -Extraction Paths
  PAC7GS

.Output file:
 -Summary passed on to printing program
  PAC7ED
 -Temporary generated source
  PAC7GP

.Output report:
 -Execution report
  PAC7DD

.Sort file(s):
  Not assigned



EXTRACTION GENERATION:  PTEX80

.Permenant input file:
 -Fixed skeleton file
  PAC7SF

.Input file:
 -Source file generated by PTEX30
  PAC7GP

.Output file:
 -Generated source to be translated
  PAC7ST


PREPROCESSOR: PAFP10
```

```
.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error message file
  PAC7AE

.Input file:
 -Generated programs
  PAF80

.Output files:
 -Generated programs to be compiled
  COB80

.Output report:
 -Execution report
  PAFREP



PTEX PRINTING: PTEXD0

.Input files:
 -VA Pac error messages


  PAC7AE
 -PTEX30 report
  PAC7ED

.Permanent input/output file:
 -Extraction Paths
  PAC7GS

.Output report:
 -Validation report
  PAC7RD

.Sort file(s):
 Not assigned
```

3.1.4. XPAF: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) XPAF BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        XPAF 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 : VALIDATION OF EXTRACTION MASTER PATH
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * . ONE '*' LINE WITH USER CODE, PASSWORD, LIBRARY AND SESSION
# * . ONE COMMAND LINE 'EX'
# *   COL 1     : ACTION CODE (ONLY WITH 'UPDATE')
# *   COL 2-3   : 'EX' CODE LINE
# *   COL 4-5   : USER ENTITY OCCURENCE CODE (7E BY DEFAUT)
# *   COL 6-11  : USER ENTITY OCCURENCE CODE
# *   COL 12-14 : LIBRARY CODE IF ENTITIES ARE IN DIFFERENT
# *             : SUBNETWORK
# *   COL 15-19 : SESSION NUMBER AND VERSION IF ENTITIES ARE IN
# *             : DIFFERENT SUBNETWORK
# *   COL 20-25 : 'UPDATE' UPDATE OF GS FILE
# *
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7GS.ini
. $PACDIR/assign/$1/SQUEL.ini
PAC7MB=$PACINPUT'MBXPAF'
export PAC7MB
PAC7ED=$PACTMP'ED'
export PAC7ED
PAC7GP=$PACTMP'GP'
export PAC7GP
PAC7DD=$PACTMP'XPAFDD.X30'
export PAC7DD
echo "Execution : PTEX30"
rtscgi PTEX30
RETURN=$?
case $RETURN in
0)
 PAC7GP=$PACTMP'GP'
 export PAC7GP
 PAC7ST=$PACTMP'ST'
 export PAC7ST
 echo "Execution : PTEX80"
 rtscgi PTEX80
 RETURN=$?
 case $RETURN in
 0)
  . $PACDIR/assign/$1/PAC7AE.ini
  . $PACDIR/assign/$1/PAC7AN.ini
  . $PACDIR/assign/$1/PAC7AR.ini
  PAF80=$PACTMP'ST'
  export PAF80
  COB80=$PACTMP'XPAF.CBL'
  export COB80
  PAFREP=$PACTMP'PAFREP'
  export PAFREP
  echo "Execution : PAFP10"
  rtscgi PAFP10
  RETURN=$?
  case $RETURN in
  0)
   . $PACDIR/assign/$1/PAC7AE.ini
   . $PACDIR/assign/$1/PAC7GS.ini
   PAC7ED=$PACTMP'ED'
   export PAC7ED
   PAC7RD=$PACTMP'XPAFRD.XD0'
   export PAC7RD
   echo "Execution : PTEXD0"
   rtscgi PTEXD0
   RETURN=$?
   case $RETURN in
   0)
    echo "End of procedure"
    echo ""
    echo "Deletion of the temporary files"
    rm -f $PACTMP'ED'
    rm -f $PACTMP'GP'
    rm -f $PACTMP'ST'
    ;;
   *)
    echo "Error in executing PTEXD0"
    ;;
   esac
   ;;
  *)
   echo "Error in executing PAFP10"
   ;;
  esac
  ;;
 *)
  echo "Error in executing PTEX80"
  ;;
 esac
 ;;
*)
 echo "Error in executing PTEX30"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

3.2. XPDM: MASTER OUTLINE

3.2.1. XPDM: INTRODUCTION

XPDM: INTRODUCTION

PRINCIPLES

A Master Outline is a P-type Volume ('V' entity) designed to be called in another PDM Volume. Its functions are to:

- Memorize general descriptions (print option, for example) so that they do not have to be redefined in each Volume.

- Print the information extracted via an Extraction Master Path. This function may be recursive.

If no serious error is detected, the XPDM procedure updates the Extraction Master Path file (GS). It can also be used without updating the GS file.

EXECUTION CONDITIONS

In order to define a Master Outline, the user must have at least a level 2 authorization.

ABNORMAL EXECUTION

For any type of abnormal end the procedure can be re-executed once the problem has been solved.

PRINTED OUTPUT

This procedure prints the description of a Master Outline, as well as the comments, and a list of the anomalies found, if any.

3.2.2. XPDM: USER INPUT

XPDM: USER INPUT

One '*' line to define the context.

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	nnnn	Session number
26	1	T	Session version
68	1	' '	Standard print
		'1'	Uppercase print

One 'EP' command line for the following elements:

POS.	LEN.	VALUE	MEANING
2	2	'EP'	Line code
4	6	rrrrrr	Report code
10	6	'UPDATE'	GS file update
		or	
		SPACE	Check of the volume's presence in GS Check of the volume's use in the sub-network. No GS file update if presence or use.

EXAMPLES

```
*user    passwordBIB
EPMANUELUPDATE

*user    passwordBIB
EPMANUEL
```

3.2.3. XPDM: DESCRIPTION OF STEPS

XPDM: DESCRIPTION OF STEPS

```
EXTRACTION OF MASTER OUTLINE:  PTED30

.Input files:
 -Error-message file
  PAC7AE
 -Index file
  PAC7AN
 -Data file
  PAC7AR

.Input transaction file:
 -User input
  PAC7MB

.Permanent input/output file:
 -Extraction paths
  PAC7GS

.Output files:
 -Report passed on to printing program
  PAC7ED
 -GS-update preparation
  PAC7SG

.Output report:
 -Execution report
  PAC7DD

GS UPDATE AND PRINTING OF THE MASTER OUTLINE:  PTED60

.Input files:
 -VA Pac error messages
  PAC7AE
 -Print file
  PAC7ED
 -GS-update preparation
  PAC7SG

.Permanent output file:
 -Extraction Paths
  PAC7GS

.Output report:
 -Execution report
  ETATGP

.Sort file(s):
 Not assigned
```

3.2.4. XPDM: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) XPDM BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        XPDM 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 : VALIDATION OF MASTER OUTLINE
# ****************************************************************
# * INPUT TRANSACTION FORMAT :
# * . ONE '*' LINE WITH USER CODE, PASSWORD, LIBRARY AND SESSION
# * . ONE COMMAND LINE 'EP'
# *  COL  1    : ACTION CODE (ONLY WITH 'UPDATE')
# *  COL  2-3  : 'EP' CODE LINE
# *  COL  4-9  : REPORT CODE
# *  COL 10-15 : 'UPDATE' UPDATE OF GS FILE
# *
# ****************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7GS.ini
PAC7MB=$PACINPUT'MBXPDM'
export PAC7MB
PAC7ED=$PACTMP'ED'
export PAC7ED
PAC7SG=$PACTMP'SG'
export PAC7SG
PAC7DD=$PACTMP'XPDMDD.D30'
export PAC7DD
echo "Execution : PTED30"
rtscgi PTED30
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7GS.ini
 PAC7ED=$PACTMP'ED'
 export PAC7ED
 PAC7SG=$PACTMP'SG'
```

```
  export PAC7SG
  ETATGP=$PACTMP'XPDMGP.D60'
  export ETATGP
  echo "Execution : PTED60"
  rtscgi PTED60
  RETURN=$?
  case $RETURN in
  0)
   echo "End of procedure"
   echo ""
   echo "Deletion of the temporary files"
   rm -f $PACTMP'ED'
   rm -f $PACTMP'SG'
   ;;
  *)
   echo "Error in executing PTED60"
   ;;
  esac
  ;;
 *)
  echo "Error in executing PTED30"
  ;;
 esac
 if [ "$RETURN" != '0' ]
 then
    sh $PACDIR/batch/proc/ERRPAUSE.ini
 fi
 exit $RETURN
```

3.3. PRGS: PRINTING OF MASTER PATH / OUTLINE FILE

3.3.1. PRGS: INTRODUCTION

PRGS: INTRODUCTION

PRINCIPLE

The PRGS procedure prints the contents of the PAC7GS file, where Master Outlines and Extraction Master Paths are stored.

PREREQUISITE

To request the printing of the Master Outline and Extraction Master Path file, the user must have at least the authorization level 2.

RESULT

A printout showing the Extraction Master Path and the associated Master Outlines.

3.3.2. PRGS: USER INPUT

PRGS: USER INPUT

One '*' line to identify the user.

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

3.3.3. PRGS: DESCRIPTION OF STEPS

PRGS: DESCRIPTION OF STEPS

```
PRINTING OF THE MASTER PATH AND OUTLINE FILE:

.Input files:
 -Error-message file
  PAC7AE
 -Extraction paths
  PAC7GS

.Input transaction file:
 -User input
  PAC7MB

.Output report:
 -Execution report
  PAC7DD
 -Master Path and Outline file report
  ETATGS

.Sort file(s):
 Not assigned
```

3.3.4. PRGS: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) PRGS BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        PRGS 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 : PRINTING FILE OF MASTER PATH
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7GS.ini
PAC7DD=$PACTMP'PRGSDD.P90'
export PAC7DD
PAC7MB=$PACINPUT'MBPRGS'
export PAC7MB
ETATGS=$PACTMP'PRGSGS.P90'
export ETATGS
echo "Execution : PTEP90"
cobrun PTEP90
RETURN=$?
case $RETURN in
0)
 ;;
8)
 echo "Error in executing PTEP90"
 echo "Error 8 : Unauthorized user"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PTEP90"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
exit $RETURN
```

4. QUALITY ANALYSIS AND CONTROL

4.1. ACTI: JOURNAL STATISTICS UTILITY

4.1.1. ACTI: INTRODUCTION

ACTI: INTRODUCTION

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

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

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

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

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

- LIBRARY CODE
- USER CODE
- ENTITY TYPE
- ENTITY CODE
- LINE CODE
- TRANSACTION TYPE (C,M,D)
- DATE OF UPDATE
- SESSION NUMBER OF UPDATE

These criteria are used to specify the REQUEST AREA.

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

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

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

EXECUTION CONDITIONS

None.

Batch procedure access authorization:
. Level 3 is required.

4.1.2. ACTI: COMMAND LANGUAGE

COMMAND LANGUAGE

COMMAND LANGUAGE

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

```
- OUTPUT    :   Output Report Type,
- PAGE      :   Page Layout (page breaks),
- AREA      :   Request Area,
- LINE      :   Statistical Report Lines,
- COLUMN    :   Statistical Report Columns,
- ABSCISSA  :   Curve-type graph Abscissas,
- ORDINATE  :   Curve-type graph Ordinates.
```

The meaning of the keywords, the parameters which define them, as well as their compatibility are explained in paragraph 'KEYWORDS DEFINITION AND VALUES'.

The OUTPUT line is required; the PAGE and AREA lines are op- tional. The LINE, COLUMN, ABSCISSA, and ORDINATE lines are either required or prohibited, depending on the requested output report type.

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

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

Request lines must be entered in the following order:

OUTPUT PAGE AREA LINE COLUMN ABSCISSA ORDINATE

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

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

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

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

PARAMETERS

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

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

Parameters which define calculations are called 'Calculations'.

The coding, meaning and compatibility of the parameters are described in paragraph 'PARAMETERS: DEFINITON AND COMMENTS'.

SEPARATORS

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

:	End of keyword,
=	Link between a parameter and its value,
( )	Set of parameters for calculations,
,	Parameter or calculation separator,
/	Calculation combination,
*	Generic selection,
Blank	End of line (subsequent data is entered for documentary purposes).

KEYWORDS MEANING AND FILLING MODES

OUT(put) OUTPUT REPORT TYPE

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

The parameters used to define the output report type are:

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

PAG(es) PAGE LAYOUT

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

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

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

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

ARE(a) REQUEST AREA

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

The REQUEST AREA line is optional.

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

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

LIN(es) DATA SORTING MODE
or
ABS(cissa)

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

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

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

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

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

COL(umns) ACTIVITY CALCULATION MODE
or
ORD(inate)

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

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

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

A printing character (&CHAR='X') must be specified for each curve.

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

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

PARAMETERS: DEFINITION AND COMMENTS

`&LIB   LIBRARY CODE`

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

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

`&USER  USER CODE`

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

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

`&ENTG  ENTITY TYPE`

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

`&ENTD  LINE CODE / ENTITY TYPE`

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

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

`&LICO  LINE CODE`

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

Values are selected according to the batch line codes.

&ENT ENTITY CODE

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

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

Values are selected according to the entity type and code.

&INPT INPUT TYPE

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

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

&D1 STARTING DATE

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

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

&D2 END DATE

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

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

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

`&S1    STARTING SESSION`

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

This parameter has to be followed by a four-character session number. If this parameter is missing, the starting session coincides with the beginning of the Journal.

`&S2   FINAL SESSION`

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

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

`&DAY  DAY-BY-DAY PRESENTATION`

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

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

`&WEEK  WEEK-BY-WEEK PRESENTATION`

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

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

&MON MONTH-BY-MONTH PRESENTATION

Used as a presentation criterion to define the page layout and the data sorting mode.
To define an X-axis, this parameter must be followed by the '=' character and the number of characters corresponding to the curve step (its default value is one character).

&YEAR YEAR-BY-YEAR PRESENTATION

Used as a presentation criterion to define the page layout and the data sorting mode.
To define an X-axis, this parameter must be followed by the '=' character and the number of characters corresponding to the curve step (its default value is one character).

&SESS PRESENTATION BY SESSION

Used as a presentation criterion to define the page layout and the data sorting mode.
The user cannot use it to select sessions (the '=' character is therefore unnecessary).

&CHAR PRINTING CURVE CHARACTER

May only be used to define the activity calculation mode relative to the curve-type graphs.
It must follow (within parentheses) the calculation defining a curve.

&INTR NUMBER OF INPUT TRANSACTIONS

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

&RETR NUMBER OF REAL TRANSACTIONS

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

PARAMETER	AREa	PAGe	OUTput	
			STA	GRA
			LIN COL	ABS ORD
&LIB	YES	YES	YES	YES
&USER	YES	YES	YES	YES
&ENTG	YES	YES	YES	YES
&ENTD		YES	YES	
&LICO	YES	YES	YES	YES
&ENT	YES	YES	YES	YES
&INPT	YES	YES	YES	YES
&D1=				
MMDDCCYY	YES		YES	YES
&D2=				
MMDDCCYY	YES		YES	YES
&S1=SESS	YES		YES	YES
&S2=SESS	YES		YES	YES
&DAY	YES	YES	YES	=
&WEEK	YES	YES	YES	=
&MON	YES	YES	YES	=
&YEAR	YES	YES	YES	=
&SESS		YES	YES	
&CHAR				CALCULATION
&INTR				CALCULATION
&RETR				CALCULATION

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

CALCULATION : only used in the Activity Calculation Mode.

The following paragraphs present some of the restrictions concerning the way requests for Journal statistics may be formulated.

GRAPHS

Page layout:

Only one parameter corresponding to a period of time may be selected (&DAY, &WEEK, &MON, &YEAR).

Data sorting mode:

Only the parameters corresponding to a Presentation period (&DAY, &WEEK, &MON, &YEAR) or to a Selection period (&D1, &D2) may be selected.

Curves:

The '*' character is used to represent the intersection point of different curves. It is therefore not desirable to use this character as a printing character for a curve. Although the user may describe up to seven curves on the same graph, it might be difficult to read the graph because of the numerous intersection points.

STATISTICAL REPORTS

Page layout:

Parameters used at this level cannot be used again to define the Data Sorting Mode.

Data sorting mode:

A selection by date following several criteria only applies to the criterion entered just before the selection. It is not possible to indicate more than one interval of the same type for a selection.

TRANSACTION LISTS

Page layout:

In the absence of page layout criteria, the transactions are presented by:

- library,
- input date,
- session number,
- user code.

The following paragraphs list the error messages going with the translation of the request in current language.

```
ERROR MESSAGES: COMMENTS

UNIDENTIFIED LINE
    The keyword identifying the line is invalid.

ABSENCE OF OUTPUT IDENTIFICATION
    The line identifying the requested report is missing.

TOO MANY REQUESTS, THE FIRST TEN ARE PROCESSED

LINES-COLUMNS INVALID WITH LISTS
    Lines, columns,  abscissas and ordinates must not appear
    on a list request.

UNKNOWN KEYWORD
    A keyword can only  be used to specify the output report
    type.

INVALID OUTPUT IDENTIFICATION

UNKNOWN PARAMETER

INVALID USE OF THE PARAMETER

NO SELECTION ALLOWED FOR THIS PARAMETER

NO SELECTION ALLOWED ON THIS LINE

TOO MANY SELECTIONS - LIMITED TO THE MAXIMUM

STEP OF THE ABSCISSA NON-NUMERIC

END DATE PRECEDES STARTING DATE

FINAL SESSION PRECEDES STARTING SESSION

INVALID OR INCOMPLETE STRUCTURE OF THE REQUEST
    Absence of lines or columns for a statistical report, or
    of abscissas or ordinates for a curve-type graph.
```

ONLY ONE ABSCISSA POSSIBLE

All the curves of the same graph must have the same abscissa.

TOO MANY COLUMNS (OR CURVES), 7 ARE PROCESSED

INVALID AGGREGATE OF TRANSACTIONS

The ordinate of a curve must be defined by a single calculation.

INVALID GRAPHIC LINE

The X-axis must be defined by a parameter corresponding to a period of time.

INVALID GRAPHIC LINE WITH PAGINATION

The period used to define the X-axis must be shorter than the one used for the page layout.

ABSENCE OF THE PRINTING CHARACTER OF THE CURVE

ONE TIME PERIOD LIMITATION FOR GRAPH PRESENTATION

The combination of several time periods is impossible for the graph page layout.

INVALID DATE

TOO MANY PRESENTATION PARAMETERS

Only 3 page layout criteria are taken into account.

PARAMETER ALSO USED AS PAGINATION

The same parameter cannot be used to define both the page layout and the data sorting mode.

4.1.3. ACTI: USER INPUT

ACTI: USER INPUT

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

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

4.1.4. ACTI: DESCRIPTION OF STEPS

ACTI: DESCRIPTION OF STEPS

```
EXTRACTION: PTU630

.Permanent input files:
 -Error message file
  PAC7AE
 -Journal Backup File
  PAC7PJ

.Transaction file:
 -Update transactions
  PAC7MB

.Output file
 -Transactions for selected reports
  PAC7ST

.Output report:
 -Batch-procedure authorization option
  PAC7DD


  - 0:  OK
  - 8:  Unauthorized user
  -12:  System error

PRINTING OF RESULTS:  PTU640

.Permanent input file:
 -Error Messages
  PAC7AE

.Input file:
 -Transactions for selected reports
  PAC7ST

.Output report:
 -Selected reports
  PAC7IV

.Sort file(s):
 Not assigned
```

4.1.5. ACTI: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) ACTI BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "----------------------------------------------------------"
echo "                        ACTI 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 : JOURNAL STATISTICS
# *********************************************************
# * INPUT TRANSACTION FORMAT :
# * .BATCH PROCEDURE ACCESS AUTHORIZATION OPTION
# *  '*' LINE WITH USER CODE AND PASSWORD
# * .ENTRIES SPECIFIC TO THE PROCEDURE
# *  SEE THE REFERENCE MANUAL "OPTIONAL UTILITIES"
# *********************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBACTI'
export PAC7MB
. $PACDIR/assign/$1/PACSAVPJ.ini
PAC7PJ=$PACSAVPJ
export PAC7PJ
PAC7ST=$PACTMP'ST'
export PAC7ST
PAC7DD=$PACTMP'ACTIDD.630'
export PAC7DD
echo "Execution : PTU630"
rtscgi PTU630
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 PAC7ST=$PACTMP'ST'
 export PAC7ST
 PAC7IV=$PACTMP'ACTIIV.640'
 export PAC7IV
 echo "Execution : PTU640"
 rtscgi PTU640
 RETURN=$?
 case $RETURN in
 0)
```

```
       echo "End of procedure"
       echo ""
       echo "Deletion of the temporary files"
       rm -f $PACTMP'ST'
       ;;
      *)
       echo "Error in executing PTU640"
       ;;
      esac
      ;;
     12)
      echo "Error in executing PTU630"
      echo "Error 12 : Error systeme"
      ;;
     8)
      echo "Error in executing PTU630"
      echo "Error 8 : Error on * input line"
      ;;
     *)
      echo "Error in executing PTU630"
      ;;
     esac
     if [ "$RETURN" != '0' ]
     then
        sh $PACDIR/batch/proc/ERRPAUSE.ini
     fi
     exit $RETURN
```

4.2. PQC-: PACBENCH QUALITY CONTROL

4.2.1. PQC: INTRODUCTION

PQC: INTRODUCTION

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

The Pacbench Quality Control facility is divided into two components:

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

- The Quality rule extraction component, customized by the user.

Two purchase options are therefore available:

- A basic option providing standard rules for quality control;

- A quality rule CUSTOMIZATION option.

The components supplied on the installation tape are:

- For both purchase options:

  . A Batch Quality Analysis procedure (PQCA);

  . A set of 'compiled' standard quality rules, in the form of a sequential file (see the Environment & Installation manual).

- For the CUSTOMIZATION option:

  . A batch procedure for the extraction and 'compilation' of the customized rules (PQCE);

  . A data element dictionary and the user entity needed for the customization of the rules, in the form of Batch transactions that the user enters in his/her own dictionary via a Batch update (UPDT). (See the Environment & Installation manual.)

4.2.2. PQCA: QUALITY ANALYSIS

4.2.2.1. PQCA: INTRODUCTION

PQCA: PACBENCH QUALITY CONTROL - ANALYSIS

PQCA: INTRODUCTION

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

CHARACTERISTICS

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

Its operation is identical to that of the standard GPRT generation-print procedure.

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

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

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

After two general programs (PACA10 and PACA20), common to all the chains, have been executed, the sub-chains are activated, according to the generation-print requests, in the following order:

- Screens
- Programs
- Dictionary

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

Once these sub-chains have been activated for the extraction of the entities to be analyzed, the PTUQ20 program performs the analysis according to the rules that it has been assigned and to the analysis parameters.

Results are printed by the PTUQ24, PTUQ25 and PTUQ30 programs.

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

EXECUTION CONDITIONS

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

USER INPUT

Please refer to the PQC Reference Manual.

OUTPUT REPORT

The user can choose between two types of reports:

. A global report showing the general results;

. A detailed report including:

- Results by entity
- Results by entity type.

The information contained in this report may also be gathered in files that will be processed by user programs. These files are:

- PACQMK for results by entity,
- PACQMJ for results by entity type.

These files are described in the PQC Reference Manual.

The procedure also prints the descriptions of the Quality-Controlled occurrences and an execution report.

PROCESSING OF THE GENERATED FLOW

This processing is identical to that of the GPRT procedure (See the corresponding chapter in this manual).

4.2.3. PQCA: DESCRIPTION OF STEPS

PQCA: DESCRIPTION OF STEPS

```
QUALITY ANALYSIS:  PACQ

The general characteristics of this step are described in
the previous sub-chapter.

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Printing command file
  PAC7AG
 -PEI environment file ('Batch')
  PAC7AB
 -PEI environment file ('on-line')
  PAC7AC
 -Error-message file
  PAC7AE
 -User parameters
  PAC7AP
 -QUALITY RULES file
  PACQMF
 -Batch-language generation skeleton
  PAC7SC
 -Dialog generation skeleton
  PAC7SG
 -Map skeleton
  PAC7SS

.Transaction files:
 -Entities to be analyzed (input)
  PAC7ME
 -Selection parameters (input)
  PACQMC

.Output reports:
 -PACQ execution report
  PAC7IA
 -VisualAge Pacbase documentation
  PAC7ID
 -Selection-parameter check
  PACQIB
 -Results by entity type
  PACQIE
 -Results by entity
  PACQIF
 -List of VA Pac identifiers which exceed the limits
  of the quality identificators
  PACQIG
 -Generation report (PEI)
  PAC7IH

.Output generated flow, made of the following output:
 -DBD generated-program file
  PAC7GB
 -OLSD generated-program file
  PAC7GE
 -C/S-OLSD generated-program file
  PAC7GG
 -Batch-language generated-program file
  PAC7GP
 -PDM generated-program file
  PAC7GV
  in the temporary files directory)
```

Other files mentioned in the procedure are temporary files used in the chains (see details in the flowcharts).

```
.Sort file(s):
 Not assigned
```

4.2.4. PQCA: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) PQCA BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "----------------------------------------------------------"
echo "                        PQCA 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 : QUALITY CONTROL ANALYSIS
# **********************************************************
. $PACDIR/assign/$1/PAC7AB.ini
. $PACDIR/assign/$1/PAC7AC.ini
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AG.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AP.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/SQUEL.ini
PAC7ME=$PACINPUT'MBPQCA'
export PAC7ME
PACQMC=$PACINPUT'MCPQCA'
export PACQMC
PACQMF=$PACINPUT'MBRULE.PQC'
export PACQMF
PAC7IA=$PACTMP'PQCA.IA'
export PAC7IA
PAC7ID=$PACTMP'PQCA.ID'
export PAC7ID
PAC7IH=$PACTMP'PQCA.IH'
export PAC7IH
PACQIB=$PACTMP'PQCA.IB'
export PACQIB
PACQIE=$PACTMP'PQCA.IE'
export PACQIE
PACQIF=$PACTMP'PQCA.IF'
export PACQIF
PACQIG=$PACTMP'PQCA.IG'
export PACQIG
PACQMJ=$PACTMP'WMJ'
export PACQMJ
PACQMK=$PACTMP'WMK'
export PACQMK
PACQMM=$PACTMP'WMM'
```

```
    export PACQMM
    PACQMN=$PACTMP'WMN'
    export PACQMN
    PACQMO=$PACTMP'WMO'
    export PACQMO
    PACQMZ=$PACTMP'WMZ'
    export PACQMZ
    PAC7EE=$PACTMP'WEE'
    export PAC7EE
    PAC7EG=$PACTMP'WEG'
    export PAC7EG
    PAC7EP=$PACTMP'WEP'
    export PAC7EP
    PAC7EV=$PACTMP'WEV'
    export PAC7EV
    PAC7GB=$PACTMP'PQCA.GB'
    export PAC7GB
    PAC7GE=$PACTMP'PQCA.GE'
    export PAC7GE
    PAC7GG=$PACTMP'PQCA.GG'
    export PAC7GG
    PAC7GP=$PACTMP'PQCA.GP'
    export PAC7GP
    PAC7GV=$PACTMP'PQCA.GV'
    export PAC7GV
    PAC7JG=$PACTMP'WJG'
    export PAC7JG
    PAC7KD=$PACTMP'WKD'
    export PAC7KD
    PAC7KE=$PACTMP'WKE'
    export PAC7KE
    PAC7KG=$PACTMP'WKG'
    export PAC7KG
    PAC7KF=$PACTMP'WKF'
    export PAC7KF
    PAC7KP=$PACTMP'WKP'
    export PAC7KP
    PAC7KS=$PACTMP'WKS'
    export PAC7KS
    PAC7KU=$PACTMP'WKU'
    export PAC7KU
    PAC7KV=$PACTMP'WKV'
    export PAC7KV
    PAC7MG=$PACTMP'WMG'
    export PAC7MG
    PAC7W1=$PACTMP'W1'
    export PAC7W1
    PAC7W2=$PACTMP'W2'
    export PAC7W2
    PAC7W3=$PACTMP'W3'
    export PAC7W3
    PAC7W4=$PACTMP'W4'
    export PAC7W4
    echo "Execution : PACQ"
    rtscgi PACQ
    echo "End of procedure"
    echo ""
    echo "Deletion of the temporary files"
    rm -f $PACTMP'W'*
    exit 0
```

4.2.5. PQCE: EXTRACTION OF USER-DEFINED QUALITY RULES

4.2.5.1. PQCE: INTRODUCTION

PQCE: EXTRACTION OF USER-DEFINED QUALITY RULES

PQCE: INTRODUCTION

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

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

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

EXECUTION CONDITIONS

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

Batch-procedure access authorization option:
. Level 2 is required.

4.2.6. PQCE: USER INPUT

PQCE: USER INPUT

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

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

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	nnnn	Session number (Blank=current session)
26	1	T	Session status if Tests session
28	1	l	Language code (F=French, A=English)
29	4	EXUE	Extractor code

For further details, see Chapter 'PACX: EXTRACTION FROM VA PAC DATABASE' in this manual.

One command line:

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

RESULT

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

PRINTED OUTPUT

This procedure prints:

1. An occurrence-extraction report

2. A check report on the validity and usage of quality indicators

3. Descriptive reports on quality rules:

- List of quality factors and criteria

- Definition and description of each indicator/metric

- Quality Control Dictionary.

4.2.7. PQCE: DESCRIPTION OF STEPS

PQCE: DESCRIPTION OF STEPS

```
EXTRACTION:  PACX

This step extracts transactions according to user input.

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE
 -Archived transactions
  PAC7PJ

.Input transaction file:
 -User input
  PAC7MB (MBPACX file in INPUT directory)

.Work files:
 -User input
  PAC7BM
 -EXPU work file
  PAC7MM
 -EXPJ work file
  PAC7MJ
 -RMEN work file
  PAC7TE
 -RMEN work file
  PAC7RE
 -RMEN work file
  PAC7RM


 -Extracted transactions
  PAC7WD
 -Multi-layered Extractor work file
  SYSEXT

.Output files:
 -Extracted transactions for UPDT
  PAC7MV (PACX.MV in the Database TMP directory)
 -Extracted transactions for REOR (EXPU)
  PAC7MR (PACX.MR in the Database TMP directory)


 -Extracted transactions for UPDP
  PAC7GY (PACX.GY in the Database TMP directory)
 -Extracted transactions for CPSN
  PAC7TD (PACX.TD in the Database TMP directory)
 -Extracted transactions for EXUE
  PAC7UE (PACX.UE in the Database TMP directory)

.Output reports:
 -General printout of the program stream
  PAC7IA
 -List of errors on input transactions
  PAC7DD
 -Summary reports on extractions


  PAC7EE
  PAC7EP
  PAC7EQ
  PAC7EZ

.Sort file(s):
  Not assigned


  0:  No error
  8:  Serious error (detailed in PAC7DD)
```

COMPILATION OF QUALITY RULES: PTUQ10

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

```
 .Permanent input file:
  -Error messages
   PAC7AE
  -Data file
   PAC7AR


 .Permanent output file:
  -'Compiled' Quality Rules
   PACQMI
```

.Transaction files:
-User input
PAC7MB
-User entity occurrences
PACQMC

.Output file:
-Preparation for printing
PACQML

.Output report(s):
-Rule-validity report
PACQIC
-Batch-procedure authorization option
PAC7DD

.Sort file(s):
Not assigned

PRINTING OF QUALITY RULES: PTUQ15

.Permanent input file:
-Error message file
PAC7AE

.Input file:
-Preparation for printing
PACQML

.Output reports:
-List of quality factors and criteria,
and description by indicator
PACQII
-Dictionary of Quality rules
PACQIJ

.Sort file(s):
Not assigned

4.2.8. PQCE: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) PQCE BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        PQCE 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 $PACDIR/batch/proc/MSGPAUSE.ini
# **************************************************************
# * VA Pac : QUALITY CONTROL - EXTRACTION
# **************************************************************
# * INPUT TRANSACTION FORMAT:
# * .. ONE '*' LINE PER USER AND LIBRARY
# * .. ONE COMMAND LINE PER ENTITY TO BE EXTRACTED
# *    COL 2-6 : 'W1EX$'
# *    COL 7   : LIBRARY SELECTION CODE
# *              'U'  ONLY SELECTED LIBRARY
# *              'C'  SELECTED LIBRARY AND HIGHER LEVEL LIB.
# *    COL 8-9 : USER ENTITY TYPE (2 CARACTERS)
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBPQCE'
export PAC7MB
PAC7BM=$PACTMP'WBM'
export PAC7BM
PAC7WD=$PACTMP'WWD'
export PAC7WD
PAC7MM=$PACTMP'WMM'
export PAC7MM
PAC7UE=$PACTMP'UE'
export PAC7UE
PAC7GY=/dev/null
export PAC7GY
PAC7TD=$PACTMP'PQCETD.PAC'
export PAC7TD
PAC7IA=$PACTMP'PQCEIA.PAC'
export PAC7IA
PAC7DD=$PACTMP'PQCEDD.PAC'
export PAC7DD
PAC7EE=$PACTMP'PQCEEE.PAC'
export PAC7EE
PAC7EZ=$PACTMP'PQCEEZ.PAC'
```

```
export PAC7EZ
PAC7EP=$PACTMP'PQCEEP.PAC'
export PAC7EP
PAC7EQ=$PACTMP'PQCEEQ.PAC'
export PAC7EQ
SYSEXT=$PACTMP'WPQCE.SY'
export SYSEXT
rm -f $SYSEXT*
PAC7PJ=/dev/null
export PAC7PJ
PAC7MV=/dev/null
export PAC7MV
PAC7TE=/dev/null
export PAC7TE
PAC7RM=/dev/null
export PAC7RM
PAC7RE=/dev/null
export PAC7RE
PAC7MR=/dev/null
export PAC7MR
PAC7MJ=/dev/null
export PAC7MJ
echo "Execution : PACX"
cobrun PACX
RETURN=$?
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/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBPQCE'
export PAC7MB
PACQMI=$PACINPUT'MBRULE.PQC'
export PACQMI
PACQMC=$PACTMP'UE'
export PACQMC
PACQML=$PACTMP'ML'
export PACQML
PACQIC=$PACTMP'PQCEIC.Q10'
export PACQIC
PAC7DD=$PACTMP'PQCEDD.Q10'
export PAC7DD
echo "Execution: PTUQ10"
```

```
cobrun PTUQ10
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUQ10"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PACQML=$PACTMP'ML'
export PACQML
PACQII=$PACTMP'PQCEII.Q15'
export PACQII
PACQIJ=$PACTMP'PQCEIJ.Q15'
export PACQIJ
echo "Execution: PTUQ15"
cobrun PTUQ15
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PTUQ15"
 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'MV'
rm -f  $PACTMP'ML'
rm -f  $PACTMP'W'*
exit $RETURN
```

5. METHODOLOGY INTEGRITY CHECK

5.1. ADM: SSADM PACDESIGN METHODOLOGY

5.1.1. SADM: INTRODUCTION

SADM: INTRODUCTION

This procedure is supplied for users of the WorkStation and the SSADM PACDESIGN application Design Methodology.

It checks the validity and the consistency of the entities that have been uploaded by the user from his/her workstation to the specifications database.

NOTE:

The SSADM methodology and the features of the SADM procedure are available only in English.

For further information, refer to the PACDESIGN Reference Manual.

EXECUTION CONDITIONS

None.

5.1.2. SADM: USER INPUT

SADM: USER INPUT

USER INPUT

One '*' line for library access:

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	nnnn	SESSION NUMBER (BLANK=CURRENT SESSION)
26	1	T	SESSION VERSION IF TEST SESSION
37	25	....	RESERVED IMS: REQUEST IDENTIFIER
			(cf. IMS BATCH PAF)

Print request lines:

POS.	LEN.	VALUE	MEANING
2	1	'T'	LINE CODE
3	1		CODE FOR REPORT TO BE PRINTED
		'V'	VALIDATION OF SSADM ENTITIES
		'1'	CROSS-BOUNDARIES DATA FLOWS WITHIN
			A DFD
		'2'	OPERATIONAL MASTERS WITHIN A DSD
		'3'	ALL ENTITIES WITH THEIR ATTRIBUTES
4	6	eeeeee	ENTITY CODE
			(required for '1' or '2')

PRINTED OUTPUT

This procedure prints the following, based on print requests:

. A validation of SSADM entities report

. List of cross-boundaries data flows within a DFD

. List of operational masters within a DSD

. List of all entities with their attributes.

5.1.3. SADM: DESCRIPTION OF STEPS

SADM: DESCRIPTION OF STEPS

```
SSADM-ENTITY CONSISTENCY CHECK:  PADM10

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE

.Transaction file:
 -User input
  PAC7MB

.Work file(s):
 -Standard PAF KSDS file
  SYSPAF

.Output report:
 -List of checked SSADM entities
  PAC7EJ
```

5.1.4. SADM: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) SADM BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        SADM 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 : VALIDATION OF PACDESIGN ENTITIES - SSADM
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * .A USER IDENTIFICATION LINE
# *  COL  2    : '*' LINE CODE
# *  COL  3-10 : USER CODE
# *  COL 11-18 : USER PASSWORD
# *  COL 19-21 : LIBRARY CODE
# *  COL 22-25 : SESSION NUMBER
# *  COL 26    : SESSION VERSION IF TEST SESSION
# * .ONE LINE PER TYPE OF PRINTING
# *  COL  2    : 'T' LINE CODE
# *  COL  3    : CODE OF THE REPORT TO BE PRINTED
# *            : 'V' VALIDATION OF SSADM ENTITIES
# *            : '1' CROSS-BOUNDARIES DATAFLOWS WITHIN A DFD
# *            : '2' OPERATIONAL MASTERS WITHIN A DSD
# *            : '3' ALL ENTITIES WITH THEIR ATTRIBUTES
# *  COL  4-9  : ENTITY CODE (IF REPORT CODE '1' OR '2')
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBSADM'
export PAC7MB
SYSPAF=$PACTMP'SYSPAF'
export SYSPAF
PAC7EJ=$PACTMP'SADMEJ.M10'
export PAC7EJ
echo "Execution : PADM10"
rtscgi PADM10
RETURN=$?
case $RETURN in
0)
```

```
  echo "End of procedure"
  echo ""
  echo "Deletion of the PAF work file"
  rm -f $PACTMP'SYSPAF'*
  ;;
 *)
  echo "Error in executing PADM10"
  ;;
 esac
 if [ "$RETURN" != '0' ]
 then
    sh $PACDIR/batch/proc/ERRPAUSE.ini
 fi
 exit $RETURN
```

5.2. YSM: WORKSTATION / YSM METHODOLOGY

5.2.1. YSMC: INTRODUCTION

YSMC: INTRODUCTION

This procedure is supplied for users of the WorkStation and the YSM Pacdesign application Methodology.

. It checks the validity and the integrity of the entities uploaded from the WorkStation to the Host Specifications Dictionary by the user.

. It checks the consistency between a Data flow Diagram and its parent diagram.

. It establishes different hierarchical lists of certain entities of the Database.

NOTE: The YSM Methodology and the procedure functionalities exist only in English.

For complete details, refer to the Pacdesign Reference Manual YSM Methodology.

EXECUTION CONDITIONS

None.

5.2.2. YSMC: USER INPUT

YSMC: USER INPUT

USER INPUT

One '*'-line for library access (required):

POS.	LEN.	VALUE	MEANING
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	User password
19	3	bbb	Code of the selected library
22	4	nnnn	Session number (space = current)
26	1	T	Session status if Test session
37	25	....	Only for IMS : Request identifier (cf. PAF batch IMS)

Entity validation request line (optional):

POS.	LEN.	VALUE	MEANING
2	1	'T'	Line code
3	1		Code of report to be printed
		'W'	'Validation of YSM entities'

PRC entity control request lines (optional):

POS.	LEN.	VALUE	MEANING
2	1	'T'	Line code
3	1		Code of report to be printed
		'Y'	'Inter process consistency checking'
4	6	eeeeee	Entity code (PRC)

Printing-request lines (optional):

POS.	LEN.	VALUE	MEANING
2	1	'T'	Line code
3	1		Code of report to be printed
		'0'	'List of Relationships'
		'4'	'Process Decomposition list (CTX)'
		'5'	'Process Decomposition list (DFD)'
		'6'	'Datastore Decomposition list'
		'7'	'Event flow Decomposition list'
		'8'	'Group Data flow Decomposition list'
		'9'	'Multiple Data flow Decomposition list'
4	6	eeeeee	Entity code (REL/CTX/PRC/DST/EFL/ DFL)

PRINTED REPORT

This procedure prints:

. A 'Validation of YSM entities' report.

. An 'Inter-process consistency check' report.

. The reports:

. 'List of relationships'.

. 'Process decomposition list (CTX)'.

. 'Process decomposition list (DFD)'.

. 'Data store decomposition list'.

. 'Event flow decomposition list'.

. 'Group Data flow Decomposition list'.

. 'Multiple Data flow Decomposition list'.

5.2.3. YSMC: DESCRIPTION OF STEPS

YSMC: DESCRIPTION OF STEPS

```
YSM METHOD INTEGRITY CHECKING:  PYSMCC

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE

.Transaction file:
 -User input
  PAC7MB

.Work file(s):
 -PAF standard KSDS file
  SYSPAF

.Output reports:
 -Integrity checking lists
  PAC7EJ
 -Validation reports
  PAC7EI


INTER-PROCESS CONSISTENCY:  PYSMC3

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error-message file
  PAC7AE

.Transaction file:
 -User input
  PAC7MB
.Work files:
 -PAF standard KSDS file
  SYSPAF

.Output report:
 -Integrity-check lists
  PAC7EJ
LIST OF RELATIONSHIPS AND REPORTS:  PYSMC2

.Permanent input files:
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Error messages
  PAC7AE

.Transaction file:
 -User input
  PAC7MB

.Work file(s):
 -PAF standard KSDS file
  SYSPAF

.Output report:
 -Integrity-check lists
  PAC7EJ
```

5.2.4. YSMC: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) YSMC BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        YSMC 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 : PACDESIGN YSM INTEGRITY CHECKING
# ***************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
PAC7MB=$PACINPUT'MBYSMC'
export PAC7MB
SYSPAF=$PACTMP'SYSPAF'
export SYSPAF
PAC7EI=$PACTMP'YSMCEI.MCC'
export PAC7EI
PAC7EJ=$PACTMP'YSMCEJ.MCC'
export PAC7EJ
echo "Execution : PYSMCC"
rtscgi PYSMCC
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7AE.ini
 . $PACDIR/assign/$1/PAC7AN.ini
 . $PACDIR/assign/$1/PAC7AR.ini
 PAC7MB=$PACINPUT'MBYSMC'
 export PAC7MB
 SYSPAF=$PACTMP'SYSPAF'
 export SYSPAF
 PAC7EJ=$PACTMP'YSMCEJ.MC3'
 export PAC7EJ
 echo "Execution : PYSMC3"
 rtscgi PYSMC3
 RETURN=$?
 case $RETURN in
 0)
  . $PACDIR/assign/$1/PAC7AE.ini
```

```
      . $PACDIR/assign/$1/PAC7AN.ini
      . $PACDIR/assign/$1/PAC7AR.ini
      PAC7MB=$PACINPUT'MBYSMC'
      export PAC7MB
      SYSPAF=$PACTMP'SYSPAF'
      export SYSPAF
      PAC7EJ=$PACTMP'YSMCEJ.MC2'
      export PAC7EJ
      echo "Execution : PYSMC2"
      rtscgi PYSMC2
      RETURN=$?
      case $RETURN in
      0)
       echo "End of procedure"
       echo ""
       echo "Deletion of working PAF file"
       rm -f $PACTMP'SYSPAF'*
       ;;
      *)
       echo "Error in executing PYSMC2"
       ;;
      esac
      ;;
     *)
      echo "Error in executing PYSMC3"
      ;;
     esac
     ;;
    *)
     echo "Error in executing PYSMCC"
     ;;
    esac
    if [ "$RETURN" != '0' ]
    then
       sh $PACDIR/batch/proc/ERRPAUSE.ini
    fi
    exit $RETURN
```

METHODOLOGY INTEGRITY CHECK	5
YSM: WORKSTATION / YSM METHODOLOGY	2
YSMC: EXECUTION JCL	4

6. PACTABLES

6.1. GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR

6.1.1. GETD - GETA: INTRODUCTION

GETD-GETA: INTRODUCTION

The TABLE DESCRIPTION GENERATOR is the interface between the Specifications Dictionary and Pactables. For further information, refer to Chapter 'GENERAL INTRODUCTION' Subchapter 'INTRODUCTION TO THE PACTABLES FACILITY' in the Pactables Reference Manual.

This interface is of interest only to users of the Pactables Facility.

This interface extracts the table descriptions necessary for Pactables from the VisualAge Pacbase Database.

This extraction is executed via either the GETA or GETD procedure according to the installation environment of the Pactables Facility:

- GETA if the Dictionary and Pactables are running under the same environment.
- GETD if the Dictionary and Pactables are running under different environments. In this case, GETD processes a table description file which is the image of the file containing the table descriptions used by the Pactables Facility. As a result, this file must be initialized before the first GETD run, by:
  - . either duplicating the description file of the Pactables Facility, if it exists,
  - . or executing the initialization procedure (GETI) described in this chapter.

GETA or GETD provide an interface file which is used as input to the GETT procedure of the Pactables Facility. For further details, refer to the Pactables Operations Manual.

EXECUTION CONDITIONS

None with regard to the specifications database, which is only read by this procedure.

Batch procedures authorization option:
.Level 2 is required.

ABNORMAL EXECUTION

If generation abends before the update of the table description file, the procedure can be restarted as it is once the error has been corrected.

If generation abends during the update of the table description file, this file must be restored before the procedure is restarted.

6.1.2. GETD - GETA: USER INPUT

GETD-GETA: USER INPUT

USER INPUT

A '*'-type line indicating the library which contains the table descriptions.

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	nnnn	Session number
26	1	t	Session status

One 'Z' line per generation or print request.

POS.	LEN.	VALUE	MEANING
2	1	'Z'	Line code
5	4		Request code:
		'TGS '	Request for table descrip. generation
		'TDS '	Request for printing of table descr.
		'TLS '	Request for list of table descriptions
		'TAS '	Request for table deletion
		'TMS '	Request for modification of frozen table characteristics
		'TGC '	Request for comments generation
9	6	ssss	Segment code of table description to be extracted ('TGS ','TGC ')
		tttttt	Table code (other requests)
15	2	' '	Not significant
17	8	DDMMCCYY	Date from which the table description can be modified. (Optional)

POS.	LEN.	VALUE	MEANING
25	8	DDMMCCYY	Date of description historical account for a G-type table. Default: last historical account.
		********	Table generation without hist. account
33	1		Data Element format type:
		' '	Internal format
		'E'	Input format
75	6	tttttt	Table number (if generating for a table other than that of the Segment's Definition file in the database).

For further information on user input, please refer to the Pactables Reference Manual.

NOTE: Table keys cannot be modified: table generation requests applying to defined tables and involving such modifications are rejected.

RESULT OBTAINED

The output of the GETA procedure is a sequential file containing table descriptions, which will be used as input to the Pactables GETT procedure.

6.1.3. GETD - GETA: DESCRIPTION OF STEPS

GETD: DESCRIPTION OF STEPS

```
EXTRACTION & UPDATE PREPARATION:  PACT40

.Permanent input files:
 -VisualAge Pacbase data file
  PAC7AR
 -VisualAge Pacbase index file
  PAC7AN
 -VisualAge Pacbase error-message file
  PAC7AE
 -Table-description file
  PAC7TD

.Input transaction file:
 -User requests
  PAC7MB

.Output report(s):
 -Transaction summary
  PAC7ET
 -Batch-procedure authorization option
  PAC7DD

.Output file:
 -Descriptions update transactions higher or equal to 2.0
  PAC7MD

.Return code:
 - 8:  Unauthorized user


FORMATTING OF DESCRIPTIONS < R 2.0:  PACT45

.Input file:
 -Description-update transactions higher or equal to 2.0
  PAC7MD

.Output file
 -Description-update transactions lower or equal to 1.2
  PAC7ND


UPDATE OF TABLE-DESCRIPTION FILE:  PACT50

(GETD procedure only)

.Permanent input file:
 -Table-description file
  PAC7TD
```

```
.Input transaction files:
 -User requests
  PAC7MB
 -Update transactions
  PAC7MD

.Output report:
 -Update review
  PAC7ET

.Sort file(s):
 Not assigned
```

6.1.4. GETD: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GETD BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        GETD 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 : TABLE DESCRIPTION GENERATOR
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE "*" LINE FOR LIBRARY FROM WHICH THE TABLE DESCRIPTIONS
# * .ONE 'Z' LINE PER GENERATION REQUEST
# *  COL  2    : 'Z' LINE CODE
# *  COL  5-8  : REQUEST CODE
# *            : 'TGS ' REQUEST FOR TABLE DESCRIP. GENERATION
# *            : 'TDS ' REQUEST FOR PRINTING OF TABLE DESCRIP.
# *            : 'TLS ' REQUEST FOR LIST OF TABLE DESCRIPTIONS
# *            : 'TAS ' REQUEST FOR DELETION OF A TABLE
# *            : 'TMS ' REQUEST FOR MODIFICATION OF A FROZEN
# *            :        TABLE CHARACTERISTICS
# *            : 'TGC ' REQUEST FOR COMMENT GENERATION
# *  COL  9-12 : SEGMENT CODE OF TABLE DESCRIPTION TO BE
# *            : EXTRACTED ('TGS ', 'TGC ')
# *  COL 13-16 : TABLE CODE (OTHERS REQUESTS)
# *  COL 17-22 : DATE FROM WHICH THE DESCRIPTION CAN BE
# *            : MODIFIED (DDMMYY)
# *  COL 23-28 : DATE OF DESCRIP. (HISTORICAL) (DDMMYY)
# *  COL 29    : DATA ELEMENT FORMAT TYPE
# *            : ' ' INTERNAL FORMAT
# *            : 'E' INPUT FORMAT
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7TD.ini
if [ ! -f "$PAC7TD" ]
then
 echo "The $PAC7TD file does not exist,"
 echo "use GETI procedure"
 exit 1
```

```
fi
PAC7MB=$PACINPUT'MBGETD'
export PAC7MB
PAC7MD=$PACINPUT'MVGETD'
export PAC7MD
PAC7ET=$PACTMP'GETDET.T40'
export PAC7ET
PAC7DD=$PACTMP'GETDDD.T40'
export PAC7DD
echo "Execution : PACT40"
rtscgi PACT40
RETURN=$?
case $RETURN in
0)
 . $PACDIR/assign/$1/PAC7TD.ini
 PAC7MB=$PACINPUT'MBGETD'
 export PAC7MB
 PAC7MD=$PACINPUT'MVGETD'
 export PAC7MD
 PAC7ET=$PACTMP'GETDET.T50'
 export PAC7ET
 echo "Execution : PACT50"
 rtscgi PACT50
 RETURN=$?
 case $RETURN in
 0)
  echo "End of procedure"
  ;;
 *)
  echo "Error in executing PACT50"
  ;;
 esac
 ;;
8)
 echo "Error in executing PACT40"
 echo "Error 8 : Error on * input line"
 ;;
*)
 echo "Error in executing PACT40"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

6.1.5. GETA: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GETA BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        GETA 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 : TABLE DESCRIPTION GENERATOR
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE "*" LINE FOR LIBRARY FROM WHICH THE TABLE DESCRIPTION
# * .ONE 'Z' LINE PER GENERATION REQUEST
# *  COL  2    : 'Z' LINE CODE
# *  COL  5-8  : REQUEST CODE
# *            : 'TGS ' REQUEST FOR TABLE DESCRIP. GENERATION
# *            : 'TDS ' REQUEST FOR PRINTING OF TABLE DESCRIP.
# *            : 'TLS ' REQUEST FOR LIST OF TABLE DESCRIPTIONS
# *            : 'TAS ' REQUEST FOR DELETION OF A TABLE
# *            : 'TMS ' REQUEST FOR MODIFICATION OF A FROZEN
# *            :        TABLE CHARACTERISTICS
# *            : 'TGC ' REQUEST FOR COMMENT GENERATION
# *  COL  9-12 : SEGMENT CODE OF TABLE DESCRIPTION TO BE
# *            : EXTRACTED ('TGS ', 'TGC ')
# *  COL 13-16 : TABLE CODE (OTHERS REQUESTS)
# *  COL 17-22 : DATE FROM WHICH THE DESCRIPTION CAN BE
# *            : MODIFIED (DDMMYY)
# *  COL 23-28 : DATE OF DESCRIP. (HISTORICAL) (DDMMYY)
# *  COL 29    : DATA ELEMENT FORMAT TYPE
# *            : ' ' INTERNAL FORMAT
# *            : 'E' INPUT FORMAT
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7TD.ini
if [ ! -f "$PAC7TD" ]
then
 echo "The $PAC7TD file does not exist,"
 echo "use GETI procedure"
 exit 1
```

```
fi
PAC7MB=$PACINPUT'MBGETA'
export PAC7MB
PAC7MD=$PACINPUT'MVGETA'
export PAC7MD
PAC7ET=$PACTMP'GETAET.T40'
export PAC7ET
PAC7DD=$PACTMP'GETADD.T40'
export PAC7DD
echo "Execution: PACT40"
rtscgi PACT40
RETURN=$?
case $RETURN in
0)
 ;;
8)
 echo "Error in executing PACT40"
 echo "Error 8: Unauthorized user"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PACT40"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
PAC7ND=$PACINPUT'NDGETA'
export PAC7ND
PAC7MD=$PACINPUT'MVGETA'
export PAC7MD
echo "Execution: PACT45"
cobrun PACT45
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PACT45"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
echo "End of procedure"
exit $RETURN
```

6.1.6. GET2 - GET1: INTRODUCTION

GET2-GET1: INTRODUCTION

GET1 and GET2 replace the GETA and GETD procedures for the generation of table-descriptions when the Pactables and VisualAge Pacbase releases are different (Pactables release 1.2 used with VisualAge Pacbase release 2.0 or higher). GET1 is the equivalent of GETA, while GET2 is the equivalent of GETD.

Use of these procedures is subject to licensed use of the Pactables Facility.

The purpose of GET1 and GET2 is to extract from the Database the table descriptions that are required for the operation of the Pactables Facility.

This extraction is performed either by GET1 or GET2, depending on the installation environment of the Pactables Facility, i.e.:

- GET1 when both the VisualAge Pacbase Repository and the Pactables Facility are in the same environment,

- GET2 if the VisualAge Pacbase Repository and the Pactables Facility are in different environments. In this case, the procedure operates with a table-description file which is an image of the description file used by the Pactables Facility.

Therefore, before running this procedure for the first time, the Table-Description file must be initialized in one of the following ways:

. Either by copying the Pactables' Table-Description file if it exists,

. Or by running the GET0 initialization procedure described in this chapter.

GET1 and GET2 produce an 'interface' file which must then be used as input to the GETT procedure of the Pactables Function. (See the Pactables Operations Manual for further information.)

EXECUTION CONDITIONS

None as far as the Specifications Database is concerned, since the procedure only reads the Database.

Option 'Batch-procedure Access Authorization':
. Authorization level 2 required.

ABNORMAL EXECUTION

If the generation process terminates unexpectedly before the start of the Description-file update, the procedure may be restarted as it is, after correction of the error that caused the abnormal ending.

If the generation terminates abnormaly while the Table-Description file is being updated, the file must be restored before the procedure can be restarted.

6.1.7. GET2 - GET1: USER INPUT

GET2-GET1: USER INPUT

USER INPUT

One '*'-line specifying the library where the Table-descriptions are stored:

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code
22	4	nnnn	Session number
26	1	t	Session status

One 'Z'-line for each generation or printing request:

Pos.	Len.	Value	Meaning
2	1	'Z'	Line code
5	4		Request code:
		'TGS '	Description-generation request
		'TDS '	Description-printing request
		'TLS '	Description-list request
		'TAS '	Table-deletion request
		'TMS '	Frozen-table characteristics modification request
		'TGC '	Comments-generation request
9	6	ssss	Segment code of table description to be extracted ('TGS', 'TGC')
		tttttt	Table code (other requests)
15	2	' '	Not significant
17	6	DDMMYY	Date from which the table description can be modified (optional)

Pos.	Len.	Value	Meaning
23	6	DDMMYY	Date of description historical account for a G-type table. Default: last historical account
		******	Generation of a table without historical account
29	1		Data-Element format type:
		' '	Internal format
		'E'	Input format
75	6	tttttt	Table number (if generating for a table other than that of the Segment Definition file in the Database)

(See the Pactables Reference Manual for further information on this input.)

NOTE: Table keys cannot be modified: table-generation requests which apply to defined tables and involve such modifications are rejected.

RESULT

The output of the GET1/GET2 procedure is a sequential file containing Table descriptions, which will be used as input for the GETT procedure of the Pactables Facility.

6.1.8. GET2 - GET1: DESCRIPTION OF STEPS

GET2: DESCRIPTION OF STEPS

```
EXTRACTION AND UPDATE PREPARATION:  PACT41

.Permanent input files:
 -VisualAge Pacbase Data file
  PAC7AR
 -VisualAge Pacbase Index file
  PAC7AN
 -VisualAge Pacbase Error-message file
  PAC7AE
 -Table-description file
  PAC7TD

.Input Transaction file:
 -Descriptions requests
  PAC7MB

.Output reports:
 -Transaction report
  PAC7ET
 -Batch-procedure authorization option
  PAC7DD

.Output file:
 -Description-update transactions lower or equal to 1.2
  PAC7MD

.Return code(s):


 -8:  unauthorized user


TABLE-DESCRIPTION UPDATE:  PACT51

(GET2 procedure only)

.Permanent input file:
 -Table-description file
  PAC7TD

.Input transaction files:
 -Descriptions requests
  PAC7MB

 -Update transactions
  PAC7MD

.Output report:
 -Update report
  PAC7ET

.Sort files:
 Not assigned.
```

6.1.9. GET2: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GET2 BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        GET2 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 : TABLE DESCRIPTION GENERATOR
# **************************************************************
# * INPUT:
# **************************************************************
# * ..ONE "*" LINE FOR THE LIBRARY FROM WHICH THE TABLE DESC
# *           TIONS WILL BE EXTRACTED.
# * ..ONE 'Z' LINE PER GENERATION REQUEST
# *   COL  2     : 'Z' LINE CODE
# *   COL  5 A 8 : REQUEST CODE
# *         'TGS ' REQUEST FOR TABLE DESCRIP. GENERATION
# *         'TDS ' REQUEST FOR PRINTING OF TABLE DESCRIP.
# *         'TLS ' REQUEST FOR LIST OF TABLE DESCRIPTIONS
# *         'TAS ' REQUEST FOR DELETION OF A TABLE
# *         'TMS ' REQUEST FOR MODIFICATION OF A FROZEN
# *                TABLE CHARACTERISTICS
# *        'TGC ' REQUEST FOR COMMENT GENERATION
# *   COL  9 A 12: SEGMENT CODE OF TABLE DESCRIPTION TO BE
# *                EXTRACTED ('TGS ', 'TGC ')
# *   COL  9 A 14: TABLE CODE (OTHERS REQUESTS)
# *   COL 17 A 22: DATE FROM WHICH THE DESCRIPTION CAN BE
# *                MODIFIED (DDMMYY)
# *   COL 23 A 28: DATE OF DESCRIP. (HISTORICAL) (DDMMYY)
# *   COL 29     : DATA ELEMENT FORMAT TYPE
# *                ' ' INTERNAL FORMAT
# *                'E' INPUT FORMAT
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7TD.ini
if [ ! -f "$PAC7TD" ]
then
 echo "The $PAC7TD file does not exist,"
 echo "use GET0 procedure"
 exit 1
fi
PAC7MB=$PACINPUT'MBGET2'
export PAC7MB
PAC7MD=$PACINPUT'MVGET2'
export PAC7MD
PAC7ET=$PACTMP'GET2ET.T41'
export PAC7ET
PAC7DD=$PACTMP'GET2DD.T41'
export PAC7DD
echo "Execution: PACT41"
rtscgi PACT41
RETURN=$?
case $RETURN in
0)
 ;;
8)
 echo "Error in executing PACT41"
 echo "Error 8: Unauthorized user"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PACT41"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7TD.ini
PAC7MB=$PACINPUT'MBGET2'
export PAC7MB
PAC7MD=$PACINPUT'MVGET2'
export PAC7MD
PAC7ET=$PACTMP'GET2ET.T51'
export PAC7ET
echo "Execution: PACT51"
rtscgi PACT51
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PACT51"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
exit $RETURN
```

6.1.10. GET1: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GET1 BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        GET1 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 : TABLE DESCRIPTION GENERATOR
# **************************************************************
# * INPUT:
# **************************************************************
# * ..ONE "*" LINE FOR THE LIBRARY FROM WHICH THE TABLE DESCR
# *           TIONS WILL BE EXTRACTED.
# * ..ONE 'Z' LINE PER GENERATION REQUEST
# *   COL  2     : 'Z' LINE CODE
# *   COL  5 A 8 : REQUEST CODE
# *         'TGS ' REQUEST FOR TABLE DESCRIP. GENERATION
# *         'TDS ' REQUEST FOR PRINTING OF TABLE DESCRIP.
# *         'TLS ' REQUEST FOR LIST OF TABLE DESCRIPTIONS
# *         'TAS ' REQUEST FOR DELETION OF A TABLE
# *         'TMS ' REQUEST FOR MODIFICATION OF A FROZEN
# *                TABLE CHARACTERISTICS
# *        'TGC ' REQUEST FOR COMMENT GENERATION
# *   COL  9 A 12: SEGMENT CODE OF TABLE DESCRIPTION TO BE
# *                EXTRACTED ('TGS ', 'TGC ')
# *   COL  9 A 14: TABLE CODE (OTHERS REQUESTS)
# *   COL 17 A 22: DATE FROM WHICH THE DESCRIPTION CAN BE
# *                MODIFIED (DDMMYY)
# *   COL 23 A 28: DATE OF DESCRIP. (HISTORICAL) (DDMMYY)
# *   COL 29     : DATA ELEMENT FORMAT TYPE
# *                ' ' INTERNAL FORMAT
# *                'E' INPUT FORMAT
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7TD.ini
if [ ! -f "$PAC7TD" ]
then
 echo "The $PAC7TD file does not exist,"
```

```
  echo "use GET0 procedure"
  exit 1
 fi
 PAC7MB=$PACINPUT'MBGET1'
 export PAC7MB
 PAC7MD=$PACINPUT'MVGET1'
 export PAC7MD
 PAC7ET=$PACTMP'GET1ET.T41'
 export PAC7ET
 PAC7DD=$PACTMP'GET1DD.T41'
 export PAC7DD
 echo "Execution: PACT41"
 rtscgi PACT41
 RETURN=$?
 case $RETURN in
 0)
  ;;
 8)
  echo "Error in executing PACT41"
  echo "Error 8: Unauthorized user"
  sh $PACDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
  ;;
 *)
  echo "Error in executing PACT41"
  echo "Error $RETURN"
  sh $PACDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
  ;;
 esac
 # ************************************************************
 echo "End of procedure"
 exit $RETURN
```

6.2. GETI-GET0: INITIALIZATION OF DESCRIPTION FILE

6.2.1. GETI: INTRODUCTION

GETI: INTRODUCTION

The GETI procedure must be executed when first using Pactables files that are stored in another environment from the VisualAge Pacbase environment. It initializes the description file in a similar way as the Pactables INTA procedure does.

6.2.2. GETI: DESCRIPTION OF STEPS

GETI: DESCRIPTION OF STEPS

INITIALIZATION OF DESCRIPTION FILE: PACTIN

.Permanent output file:
 -Table description file
  PAC7TD

.Transaction input file:
 -Parameter line
  PAC7MD

POS.	LEN.	VALUE	MEANING
1	36		Installation name
37	1		Language code:
		'F'	French (Default option)
		'E'	English
38	1		DOS only: machine date inversion
		' '	MM/DD/CCYY (Default option)
		'I'	DD/MM/CCYY
39	12		Not used
51	4	cccc	Class for security system
55	1		Type of security system
		'R'	RACF
		'S'	TOP SECRET
56	2	nn	Number of lines per printing page
58	1		Type of resource controls
		' '	Def.tables resources security system
		'P'	Def.resources in VA Pacbase
59	1		Lock of the user's code
		' '	Other user's code authorized
		'N'	Other user's code unauthorized

Output report:
-Initialization review
  PAC7ED

6.2.3. GETI: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GETI BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        GETI 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 : TABLE DESCRIPTION FILE INITIALIZATION
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE INITIALIZATION LINE
# *  COL  1-36 : INSTALLATION NAME
# *  COL 37    : LANGUAGE CODE
# *            : 'F' FRENCH (DEFAULT OPTION)
# *            : 'E' ENGLISH
# **************************************************************
. $PACDIR/assign/$1/PAC7TD.ini
PAC7MD=$PACINPUT'MBGETI'
export PAC7MD
PAC7ED=$PACTMP'GETIED.TIN'
export PAC7ED
echo "Execution : PACTIN"
rtscgi PACTIN
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
*)
 echo "Error in executing PACTIN"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

6.2.4. GET0: INTRODUCTION

GET0: INTRODUCTION

The GET0 procedure initializes the table-descriptions when the Pactables release in use is Rel. 1.2 while the VisualAge Pacbase release is Rel. 2.0 or higher. It is the equivalent of the GETI procedure.

The function of GET0 is the following:
When first using Table files that are disconnected from VA Pac, it initializes the Table-Description file in the same way as the INTA procedure of the Pactables Function.

6.2.5. GET0: DESCRIPTIION OF STEPS

GET0: DESCRIPTION OF STEPS

INITIALIZATION OF DESCRIPTION FILE: PACTI1

.Permanent output file:
 -Table-description file
  PAC7TD

.Input transaction file:
 -Parameter line
  PAC7MD

Pos.	Len.	Value	Meaning
1	36		Installation label
37	1		Language code
		'F'	French (default option)
		'E'	English
38	1		DOS only: inversion of machine-date
		' '	MM/DD/YY (default option)
		'I'	DD/MM/YY

.Output report:
 -Initialization report
  PAC7ED

6.2.6. GET0: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) GET0 BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        GET0 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 : TABLE DESCRIPTION FILE INITIALIZATION
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * .ONE INITIALIZATION LINE
# *  COL  1-36 : INSTALLATION NAME
# *  COL 37    : LANGUAGE CODE
# *            : 'F' FRENCH (DEFAULT OPTION)
# *            : 'E' ENGLISH
# **************************************************************
. $PACDIR/assign/$1/PAC7TD.ini
PAC7MD=$PACINPUT'MBGET0'
export PAC7MD
PAC7ED=$PACTMP'GET0ED.TI1'
export PAC7ED
echo "Execution : PACTI1"
rtscgi PACTI1
RETURN=$?
case $RETURN in
0)
 echo "End of procedure"
 ;;
*)
 echo "Error in executing PACTI1"
 ;;
esac
if [ "$RETURN" != '0' ]
then
   sh $PACDIR/batch/proc/ERRPAUSE.ini
fi
exit $RETURN
```

7. PAC/IMPACT

FOREWORD

```
------------------------------------------------------------
NOTE:  Pac/Impact users may also refer to the 'Pac/Impact
       for VA Pac' Reference Manual.
------------------------------------------------------------
```

Impact analysis requires a very large amount of machine-time. It is therefore recommended to limit the scope of the analysis.

You can limit your analysis to two distinct levels. You can also combine two levels, to define a more precise analysis domain.

1. The UXSR procedure, documented in Sub-Chapter 'UXSR: Partial Sub-Network Extraction', Chapter 'MANAGER'S UTILITIES' of the Batch Procedures: Administrator's Guide, allows you to create a new image of the VA Pac Database, by zooming on a given sub-network. This creates a new database which is a subset (restructured and/or renamed) of the initial database. The analysis is then performed on this subset.

   NOTE: Extraction of a session is also possible.

   Furthermore, the REOR procedure (which must always be run after a UXSR) allows you to cancel those occurrences which are not relevant to the analysis.

2. You may also choose to limit your analysis to certain occurrences of the Program, Screen or Database Block entities. Additional selection options are available to this effect.

   This analysis limitation is performed by the INFP utility, documented in the ENVIRONMENT AND INSTALLATION Manual, Chapter 'INSTALLATION', Sub-Chapter 'Initialization of the FP file', as well as in the Pac/Impact for VA Pac Reference Manual.

3. The procedures in this Function do not impact the database files. However, it is recommended to close the on-line files for better performance.

7.1. ISEP: SELECTION OF ENTRY POINTS

7.1.1. ISEP: INTRODUCTION

ISEP: INTRODUCTION

The ISEP procedure is designed to select the entry points -- Data Elements and/or character strings -- which will be used as criteria by the impact analysis (IANA procedure).

The identification line of the selection context (* line) is required. It allows you to specify the session and the sub-network (view Z1) from which the selection will be made.

Data Elements and character strings are considered as entry points when they meet selection criteria entered in ISEP user input lines (or command lines).

Three types of criteria may be used (see below) and at least one selection criterion is required, knowing that no particular criterion type is required.

A selection may combine several types of criteria, and several command lines for each type.

. The E-type line allows you to extract Data Elements by selecting a code (generic code authorized) and/or one or several format(s).

. The S-type line allows you to extract character strings by selecting a code (generic code authorized) and/or one or several format(s).

. The W-type line allows you to select Data Elements via a keyword. You may also indicate the keyword type, Data Element formats and code.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Whatever the cause of the abend, the procedure can be re-run as it is, after correction of the problem.

7.1.2. ISEP: USER INPUT

ISEP: USER INPUT

Only one '*' line (required, placed at the beginning of the stream):

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Code of the highest library in the sub-network
22	4	ssss	Session number (blank if current session)
26	1		Session status (' ' or 'T')
28	1	F or E	Language code if different from that of the site (bilingual sites only)
69	3	iii	Code of the lowest library in the sub-network (optional)

One E-type line: Selection of Data Elements (optional)

Pos.	Len.	Value	Meaning
2	1	'E'	Line code
3	6		Data Element code (generic code possible with the '*' character, at beginning or end of code: ***XXX or XXX***, or with the '?' character followed by the string to be included in the code (?XXX).
9	10		Data Element input format
19	10		Data Element internal format
29	1		Internal usage (default: D)
30	27		Data Element output format
57	1	'N'	Child Data Elements not impacted
		' '	Child Data Elements impacted

One S-type line: Selection of character strings (optional)

Pos.	Len.	Value	Meaning
2	1	'S'	Line code
3	30		String code (generic code possible with the '*' character anywhere in the code), or ?xx where xx is a string located anywhere in the sequence of char.
33	10		Internal format of the string
43	1		Internal usage (Default: D)

One W-type line: Selection on keyword (optional)

Pos.	Len.	Value	Meaning
2	1	'W'	Line code
3	1		Keyword type (implicit 'L', explicit 'M', or both ' ')
4	13		Keyword code (no generic code)
17	10		Data Element input format
27	10		Data Element internal format
37	1		Internal usage (Default: D)
38	27		Data Element output format
65	6		Data Element code (generic code possible with the '*' character anywhere in the code)
71	1	'N'	Child Data Elements not impacted
		' '	Child Data Elements impacted

7.1.3. ISEP: DESCRIPTION OF STEPS

ISEP: DESCRIPTION OF STEPS

```
SELECTION OF ENTRY POINTS:  PAN210

.Permanent input files:
 -Error messages
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -File of entities to be analyzed
  PAC7FP

.Transactions file:
 -User input
  PAC7MB

.Output file:
 -Selected entry points
  PAC7FH

.Output report(s):
 -Validation report
  PAC7IE


 .  0 : OK.
 . 12 : System error


REMOVAL OF DUPLICATE ENTRY POINTS:  PAN215

.Transactions file:
 -Selected entry points
  PAC7FH

.Permanent output files:
 -Sorted selected entry points
  PAC7HF
 -Reduced entry points to be purged
  PAC7FR

.Sort file(s):
 Not assigned

.Return codes:
  - 0:  OK
  -12:  System error
```

7.1.4. ISEP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) ISEP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        ISEP 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 : Pac/Impact - SELECTION OF ENTRY POINTS
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * . * LINE (MANDATORY)
# * COL  2     : '*' LINE CODE
# * COL  3-10  : uuuuuuuu  USER CODE
# * COL 11-18  : pppppppp  USER PASSWORD
# * COL 19-21  : bbb       LIBRARY CODE
# * COL 22-25  : ssss      SESSION NUMBER UTILE
# *            :           (BLANK IF CURRENT SESSION)
# * COL 26     : ' '       SESSION  STATUS
# *            : 'T'
# * COL 28     : 'E'       LANGUAGE CODE IF DIFFERENT FROM THAT
# *            : 'F'       OF THE SITE
# *
# * ONE E-TYPE LINE: SELECTION OF DATA ELEMENTS (OPTIONAL)
# * COL  2     : 'E' LINE CODE
# * COL  3-8   :     DATA ELEMENT CODE (GENERIC CODE POSSIBLE
# *            :     WITH THE * CHARACTER, ANYWHERE IN THE CODE)
# * COL  9-18  :     DATA ELEMENT INPUT FORMAT
# * COL 19-28  :     DATA ELEMENT INETRNAL FORMAT
# * COL 29     :     INTERNAL USAGE (DEFAULT: 'D')
# * COL 30-56  :     DATA ELEMENT OUTPUT FORMAT
# * COL 57     : 'N' CHILD DATA ELEMNTS NOT IMPACTED
# *
# *
# * .One S-type line: Selection of character strings (optional)
# * COL  2     : 'S' LINE CODE
# * COL  3-32  :     STRING CODE (GENERIC CODE POSSIBLE
# *            :     WITH THE * CHARACTERE, ANYWHERE IN THE
# *            :     CODE), OR ?XX OR XX IS A STRING LOCATED
# *            :     ANYWHERE IN THE SEQUENCE OF CHAR
# * COL 33-42  :     INTERNAL FORMAT OF THE STRING
# * COL 43     :     INTERNAL USAGE (DEFAULT: 'D')
# *
# * .ONE W-TYPE LINE: SELECTION ON KEYWORD (OPTIONAL)
# * COL  2     : 'W' LINE CODE
# * COL  3     :     KEYWORD TYPE (IMPLICIT 'L', EXPLICIT 'M'
# *            :     OR BOTH ' ')
# * COL  4-16  :     KEYWORD CODE (NO GENERIC CODE)
# * COL 17-26  :     DATA ELEMENT INPUT FORMAT
# * COL 27-36  :     DATA ELEMENT INTERNAL FORMAT
# * COL 37     :     INTERNAL USAGE (DEFAULT: D)
# * COL 38-64  :     DATA ELEMENT OUTPUT FORMAT
# * COL 65-70  :     DATA ELEMENT CODE (GENERIC CODE POSSIBLE
# *            :     WITH '*' CHARACTER ANYWHERE IN THE CODE)
# * COL 71     : 'N' CHILD DATA ELEMENTS NOT IMPACTED,
# *            :     ' ' CHILD DATA ELEMENTS IMPACTED
# ***************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7FP.ini
PAC7MB=$PACINPUT'MBISEP'
export PAC7MB
PAC7FH=$PACTMP'HF'
export PAC7FH
PAC7IE=$PACTMP'ISEPIE.210'
export PAC7IE
echo "Execution : PAN210"
cobrun PAN210
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN210"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN210"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
. $PACDIR/assign/$1/PAC7FH.ini
PAC7HF=$PAC7FH.NEW
export PAC7HF
. $PACDIR/assign/$1/PAC7FR.ini
PAC7FR=$PAC7FR.NEW
export PAC7FR
PAC7FH=$PACTMP'HF'
export PAC7FH
echo "Execution : PAN215"
cobrun PAN215
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN215"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN215"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
echo ""
echo "Call of FHBACKUP.ini file"
. $PACDIR/assign/$1/FHBACKUP.ini
echo "Call of FRBACKUP.ini file"
. $PACDIR/assign/$1/FRBACKUP.ini
echo ""
echo "Deletion of the temporary files"
rm -f  $PACTMP'HF'
exit $RETURN
```

7.2. IPEP: ENTRY-POINT PRINTOUT

7.2.1. IPEP: INTRODUCTION

IPEP: INTRODUCTION

The IPEP procedure produces two types of printouts.

1. List of entry points:

   This list is obtained after the ISEP procedure, since this procedure selects the entry points.

2. List of impact search criteria:

   This list is obtained after the IANA procedure, since this procedure selects the impact search criteria.

In the printout, the criteria or entry points are sorted by alphabetical order (Data Elements and character strings altogether) for each definition library of these criteria.

The order of printing of the categories is:

-character string
-Data Element defined in Dictionary
-Data Element defined in Segment Description
-Data Element defined in Report Structure
-Data Element defined in the Screen or Program Working Section.

EXECUTION CONDITIONS

None, but the FH file must exist.

ABNORMAL EXECUTION

Whatever the cause of the abend, the procedure can be run again as it is, after the problem has been solved.

USER INPUT

No user input is required for the execution of the IPEP procedure.

7.2.2. IPEP: DESCRIPTION OF STEPS

IPEP: DESCRIPTION OF STEPS

```
PRINTING OUT ENTRY POINTS:  PAN220

.Permanent input files:
 -Error messages
  PAC7AE
 -Entry points
  PAC7HF

.Output report:
 -List of entry points
  PAC7IL

.Sort file(s):
 Not assigned


 - 0:  OK
 -12:  System error
```

7.2.3. IPEP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) IPEP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        IPEP PROCEDURE"
echo "                        =============="
echo "Directory 'assign'                : $PACDIR/assign/$1"
. $PACDIR/assign/$1/PACTMP.ini
echo "Directory 'tmp'                   : `dirname $PACTMP.`"
echo "---------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/MSGPAUSE.ini
# **************************************************************
# * VA Pac : Pac/Impact - PRINTING ENTRY POINTS
# **************************************************************
# * NO USER INPUT
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7FH.ini
PAC7HF=$PAC7FH
export PAC7HF
PAC7IL=$PACTMP'IPEPIL.220'
export PAC7IL
echo "Execution : PAN220"
cobrun PAN220
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN220"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN220"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
exit $RETURN
```

7.3. ISOS: SELECTION OF STRINGS AND OPERATORS

7.3.1. ISOS: INTRODUCTION

ISOS: INTRODUCTION

ISOS is a complement to the ISEP procedure. Its purpose is to select the following items:

. VA Pac-processed dates, such as DATOR and DAT8, that will be used as entry points to perform the impact analysis from the first iteration (IANA procedure),

. Character-strings, without considering them as entry points (such as ORDER BY). For the strings which provide entry points, see the description of the 'S'-type line in the ISEP procedure's USER INPUT section,

. Operators used in procedural code (-P) lines, such as ADT. Some of these operators trigger the generation of date-type entry points (such as DATOR for ADT),

. Lines that use constant values, either defined (VALUE), moved (MOVE), or conditionned ('IF').

Reports on entities using these operators and character-strings can be produced on request (IPAI procedure).

NARROWING THE SELECTION SCOPE

For better performance, it is advisable to narrow the scope of the selection. This can be done at two different levels, and should always be done before running the procedure.

. Via the UXSR procedure, documented in sub-chapter 'Partial Sub-Network Extraction', you can create another VA Pac Database. The new Database is a subset (restructured and/or renamed) of the initial Database. The analysis will be performed on this subset.

. Via the INFP utility, documented in sub-chapter 'INFP : FP File Initialization (Impact Analysis)', you can decide to restrict the scope of the selection to entities of a particular type or types, or to particular entities of a given type. Further selection options are also available.

The selection context's identification line (*-line) is required. It allows you to specify, besides the session, the library from which you want to build the sub-network that will be analyzed (view Z1).

Three types of selection may be used (see below). At least one type of selection is required, no particular type being requested.

The selection may include more than one type of selection, and more than one command line for each type.

. The 'D'-type line allows you to request the extraction of date-type Data Elements handled by VisualAge Pacbase.

The maximum number of 'D'-lines is 40.

. The 'C'-type line allows you to extract character-strings that are likely to include one or more blanks. In this case, the separator must be specified, and the number of blanks is significant. These strings are not entry points.

The maximum number of 'C'-lines is 50 characters for each one of the three search domains.

. The 'O'-type line allows you to select operators processed in -P lines.

The maximum number of 'O'-lines is 50.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Whatever the cause of an abnormal ending, the procedure may be re-run as it is after correction of the problem.

7.3.2. ISOS: USER INPUT

ISOS: USER INPUT

Only one '*'-line (required, placed at the beginning of the stream):

Pos.	Len.	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Code of the highest library in the sub-network
22	4	ssss	Session number (blank if current session)
26	1		Session status (' ' or 'T')
28	1	F or E	Language code if different from that of the site (bilingual sites only)
69	3	iii	Code of the lowest library in the sub-network (optional)

One 'D'-line for the selection of generated dates (optional):

Pos.	Len.	Value	Meaning
2	1	'D'	Line code
3	9		Code of generated date Data-Element to be extracted (which must be recognized by the system)

One 'O'-line for the selection of operators (optional):

Pos.	Len.	Value	Meaning
2	1	'O'	Line code
3	3		Code of wanted operator (which must be recognized by the system)

One 'C'-line for the selection of character strings (optional):

Pos.	Len.	Value	Meaning
2	1	'C'	Line code
3	1		End-of-string separator (Required if the string contains at least one blank)
4	31		Code of searched string. (Must be ended by the separator if a separator is specified)
35	1		Where the string is to be searched:
		'D'	Search in the Definition part (-W of programs and/or screens, and -9 of programs)
		'T'	Search in Procedural Code part (-P of programs and/or screens, -8, -9, -SC of programs, -CE and -CS of screens)
		'R'	Search in Report-specific Procedural code part: .Category condition and Structure .Source Data-Element code (Struct.)
		' '	Search in the three above mentioned parts

One 'V'-line for the selection of constant values (optional):

Pos.	Len.	Value	Meaning
2	1	'V'	Line code
3	1		Beginning-of-value separator Required (either ' or ")
4	31		Code of searched value Required, ending with the separator (either ' or ")
35	1		Where the constant is to be searched
		'D'	Search in the Definition part (-W of programs and/or screens, and -9 of programs)
		'T'	Search in the Procedural Code part (-P of programs and/or screens, -8, -9, -SC of programs, -CE and -CS of screens)
		'R'	Search in Report-specific Procedural code part: .Category condition and Structure .Source Data-Element code (Struct.)
		' '	Search in the three above mentioned parts

7.3.3. ISOS: DESCRIPTION OF STEPS

ISOS: DESCRIPTION OF STEPS

```
SELECTION OF STRINGS AND OPERATORS:  PAN212

.Permanent input files:
 -Error messages
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Entities in production
  PAC7FP

.Transaction file:
 -User input
  PAC7MB

.Output file(s):
 -Selected entry points
  PAC7FH
 -Impact analysis results
  PAC7MF

.Output report(s):
 -Validation report
  PAC7IE


 - 0:  OK
 -12:  System error
```

```
DELETION OF DUPLICATE ENTRY POINTS:  PAN215

.Transaction file:
 -Selected entry points
  PAC7FH

.Permanent output files:
 -Sorted selected entry points
  PAC7HF
 -Reduced entry points to be purged
  PAC7FR

.Sort file(s):
 Not assigned.


 - 0:  OK
 -12:  System error


UPDATE OF IMPACT ANALYSIS RESULTS:  PAN260

.Transaction file:
 -Impact analysis result (for that iteration)
  PAC7MF

.Permanent input file:
 -Results from preceding analysis
  PAC7OF

.Permanent output file:
 -Sorted impact-analysis results
  PAC7FO

.Sort file(s):
  Not assigned

.Return codes:
 - 0:  OK
 -12:  System error
```

7.3.4. ISOS: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) ISOS BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        ISOS 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 : Pac/Impact - SELECTION OF STRINGS AND OPERATORS
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7FP.ini
PAC7MB=$PACINPUT'MBISOS'
export PAC7MB
PAC7FH=$PACTMP'HF'
export PAC7FH
PAC7MF=$PACTMP'MF'
export PAC7MF
PAC7IE=$PACTMP'ISOSIE.212'
export PAC7IE
echo "Execution : PAN212"
cobrun PAN212
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN212"
 echo "ERREUR 12 : System Error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN212"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
```

```
# *************************************************************
. $PACDIR/assign/$1/PAC7FH.ini
. $PACDIR/assign/$1/PAC7FR.ini
PAC7HF=$PAC7FH.NEW
export PAC7HF
PAC7FR=$PAC7FR.NEW
export PAC7FR
PAC7FH=$PACTMP'HF'
export PAC7FH
echo "Execution : PAN215"
cobrun PAN215
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN215"
 echo "ERREUR 12 : System Error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN215"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# *************************************************************
. $PACDIR/assign/$1/PAC7FO.ini
PAC7OF=$PAC7FO
export PAC7OF
PAC7FO=$PAC7FO.NEW
export PAC7FO
PAC7MF=$PACTMP'MF'
export PAC7MF
echo "Execution : PAN260"
cobrun PAN260
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PAN260"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# *************************************************************
echo "End of procedure"
echo ""
echo "Call of FHBACKUP.ini file"
. $PACDIR/assign/$1/FHBACKUP.ini
echo "Call of FOBACKUP.ini file"
. $PACDIR/assign/$1/FOBACKUP.ini
echo "Call of FRBACKUP.ini file"
. $PACDIR/assign/$1/FRBACKUP.ini
echo ""
echo "Deletion of the temporary files"
rm -f  $PACTMP'MF'
rm -f  $PACTMP'HF'
rm -f  $PACTMP'FH'
exit $RETURN
```

7.4. IMFH: MERGE OF FH FILES - CREATION OF FH AND FR

7.4.1. IMFH: INTRODUCTION

IMFH: INTRODUCTION

The IMFH procedure allows you to merge two or more FH files so as to:

- Have only one FH file, after eliminating possible duplicates;

- Obtain a FR file synchronized with the created FH file.

This procedure should be used when you want to merge the FH file produced by the ISEP procedure with that issued by the ISOS procedure.

A subsidiary use of this procedure is to recreate the FR file from a FH file.

7.4.2. IMFH: DESCRIPTION OF STEPS

IMFH: DESCRIPTION OF STEPS

```
DELETION OF DUPLICATE ENTRY POINTS:  PAN215

.Transaction file:
 -Selected entry points
  PAC7FH

.Permanent output files:
 -Sorted selected entry points
  PAC7HF
 -Reduced entry points to be purged
  PAC7FR

.Sort file(s):
 Not assigned

.Return codes:
 .  0: OK.
 . 12: system error
```

7.4.3. IMFH: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) IMFH BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        IMFH 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 : Pac/Impact - FH FILES FUSION AND FR FILE CREATION
# ***************************************************************
. $PACDIR/assign/$1/PAC7FH.ini
. $PACDIR/assign/$1/PAC7FR.ini
cat $PAC7FH > $PACTMP'FH.TMP'
cat $PAC7FH'-1' >> $PACTMP'FH.TMP'
PAC7FH=$PACTMP'FH.TMP'
export PAC7FH
PAC7HF=$PAC7FH.NEW
export PAC7HF
PAC7FR=$PAC7FR.NEW
export PAC7FR
echo "Execution : PAN215"
cobrun PAN215
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN215"
 echo "ERREUR 12 : System Error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN215"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
echo "End of procedure"
```

```
    echo ""
    echo "Call of FHBACKUP file"
    . $PACDIR/assign/$1/FHBACKUP.ini
    echo "Call of FRBACKUP file"
    . $PACDIR/assign/$1/FRBACKUP.ini
    echo ""
    echo "Deletion of the temporary files"
    rm -f $PACTMP'FH.TMP'
    exit $RETURN
```

7.5. IANA: IMPACT SEARCH CRITERIA

7.5.1. IANA: INTRODUCTION

IANA: INTRODUCTION

The IANA procedure is used to search Data Elements and character-strings according to:

1. The entry points provided by the ISEP procedure when IANA is run for the first time,
2. The impact search criteria produced by a preceding execution of IANA.

IANA is therefore an iterative process, which runs until no more impact search criteria are found.

Prior to an IANA execution, you have the choice to inhibit unwanted:

1. Entry points, after an execution of the ISEP procedure,
2. Impact search criteria, after a preceding execution of the IANA procedure.

In both cases, deletions are made in the FR file, (under an editor) either by physical deletion, or by inhibition (value 'E' in the action code of the corresponding lines).

The entry points (first iteration) or impact search criteria (further iterations) are printed once the purged criteria have been taken into account. This printout sorts criteria into 'accepted' and 'rejected' criteria. The file which contains the already impacted criteria may be reinitialized if you do not need to save them.

However, it is recommended to reinitialize this file before the first execution of IANA which follows a new execution of ISEP. To reinitialize the FQ file, run the INFQ procedure documented thereafter.

The impact analysis file may either be empty or contain the results of different execution contexts. It allows to compound the results of all iterations of the impact analysis for a given context.

The FP file used as input for the analysis procedures, contains the list of entities or entity types to be analyzed. If no user input is entered in this file before it is initialized by the INFP procedure, all analyzable entities will be analyzed.

Entities which are to be analyzed are specified in the FP file via the following coding: type coded on 3 characters, entity coded on 6 characters (****** being the generic entity code).

The impact results file may be empty, or contain the impact analysis for other execution contexts. It can accumulate the results of all the impact analysis iterations for a given context.

EXECUTION CONDITIONS

The FH file -- entry points or impact search criteria -- must exist and must not be empty.

ABNORMAL EXECUTION

Whatever the cause of the abend, you can run the procedure again as it is, after the problem has been solved.

However, the status of the FH, FR, and FO generation files should be checked.

USER INPUT

The IANA procedure does not require any specific user input.

This procedure is iterative as long as the FH file (impact search criteria) is not empty (return code set to value 4 if empty, and to value 0 otherwise).

7.5.2. IANA: DESCRIPTION OF STEPS

IANA: DESCRIPTION OF STEPS

```
RECOGNITION OF CRITERIA AFTER THE PURGE:  PAN230

.Permanent input files:
 -Search criteria
  PAC7FH
 -Criteria after purge (reduced file)
  PAC7FR

.Output file:
 -Search criteria
  PAC7HF


PRINTING OF ENTRY POINTS:  PAN220

.Permanent input files:
 -Error messages
  PAC7AE
 -Sorted criteria
  PAC7HF

.Output report(s):
 -List of accepted / rejected criteria
  PAC7IL

.Sort file(s):
 Not assigned


IMPACT ANALYSIS:  PAN250

.Permanent input files:
 -Error messages
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -File of entities to be analyzed
  PAC7FP


.Transaction file:
 -Impacted criteria
  PAC7FH

.Input-output file:
 -Impacted criteria already processed
  PAC7FQ

.Output files:
 -New impacted criteria
  PAC7HF
 -Impact analysis results
  PAC7MF

.Return codes:
 - 0:  OK
 -12:  System error



UPDATE OF IMPACT ANALYSIS RESULTS:  PAN260

.Transaction file:
 -Impact analysis results (level)
  PAC7MF

.Permanent input file:
 -Results of previous analysis
  PAC7OF

.Permanent output file:
 -Sorted results of impact analysis
  PAC7FO

.Sort file(s):
 Not assigned

.Return codes:
 - 0:  OK
 -12:  System error


REMOVAL OF DUPLICATE ENTRY POINTS:  PAN215

.Transaction file:
 -Selected entry points
  PAC7FH

.Permanent output file:
 -Sorted selected entry points
  PAC7HF
 -Reduced entry points to be purged
  PAC7FR

.Sort file(s):
 Not assigned

.Return codes:
 - 0:  OK
 -12:  System error
```

7.5.3. IANA: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) IANA BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        IANA 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 : Pac/Impact - IMPACTS ANALYSIS
# **************************************************************
# * NO USER INPUT
# **************************************************************
. $PACDIR/assign/$1/PAC7FH.ini
. $PACDIR/assign/$1/PAC7FR.ini
PAC7HF=$PACTMP'HF'
export PAC7HF
echo "Execution : PAN230"
cobrun PAN230
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN230"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN230"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7FH.ini
PAC7HF=$PAC7FH
export PAC7HF
PAC7IL=$PACTMP'IANAIL.220'
```

```
export PAC7IL
echo "Execution : PAN220"
cobrun PAN220
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN220"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN220"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7FQ.ini
. $PACDIR/assign/$1/PAC7FP.ini
cp $PAC7FQ $PAC7FQ.NEW
cp $PAC7FQ.idx $PAC7FQ.NEW.idx
PAC7FQ=$PAC7FQ.NEW
export PAC7FQ
PAC7HF=$PACTMP'FH'
export PAC7HF
PAC7MF=$PACTMP'MF'
export PAC7MF
PAC7FH=$PACTMP'HF'
export PAC7FH
echo "Execution : PAN250"
cobrun PAN250
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN250"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
. $PACDIR/assign/$1/PAC7FO.ini
PAC7OF=$PAC7FO
export PAC7OF
PAC7FO=$PAC7FO.NEW
export PAC7FO
PAC7MF=$PACTMP'MF'
export PAC7MF
echo "Execution : PAN260"
cobrun PAN260
RETURN=$?
case $RETURN in
0)
 ;;
*)
```

```
 echo "Error in executing PAN260"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
. $PACDIR/assign/$1/PAC7FH.ini
PAC7HF=$PAC7FH.NEW
export PAC7HF
. $PACDIR/assign/$1/PAC7FR.ini
PAC7FR=$PAC7FR.NEW
export PAC7FR
PAC7FH=$PACTMP'FH'
export PAC7FH
echo "Execution : PAN215"
cobrun PAN215
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PAN215"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ************************************************************
echo "End of procedure"
echo ""
echo "Call of FHBACKUP.ini file"
. $PACDIR/assign/$1/FHBACKUP.ini
echo "Call of FOBACKUP.ini file"
. $PACDIR/assign/$1/FOBACKUP.ini
echo "Call of FRBACKUP.ini file"
. $PACDIR/assign/$1/FRBACKUP.ini
echo "Call of FQBACKUP.ini file"
. $PACDIR/assign/$1/FQBACKUP.ini
echo ""
echo "Deletion of the temporary files"
rm -f  $PACTMP'MF'
rm -f  $PACTMP'HF'
rm -f  $PACTMP'FH'
exit $RETURN
```

7.6. IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS

7.6.1. IPIA: INTRODUCTION

IPIA: INTRODUCTION

The IPIA procedure is used to print Reports on the analysis results and to format these results in batch update transactions.

Possible reports produced by IPIA are the following:

1. Analysis results by entry point:

   Analysis follow-up of the subsequent iterations.

   >>> Report requested by value '1' in Position 7 of the P-type user input line.

2. List of impact search criteria by entry point:

   Valid when the IANA iteration is completed.

   >>> Report requested by value '1' in Position 8 of the P-type user input line.

3. Analysis results by Library:

   Results are formatted in batch update transactions (print or file output).

   >>> Report requested by value '1' in Position 9 of the P-type user input line.

   Additional option (page and line skips) requested by value '2' in Position 9.

   >>> File requested by value '1' in Position 12.

4. Impacted-occurrences summary:

   List of all impacted occurrences with the number of impacted lines, for each type of line, not sorted by entry points.

   >>> Report requested by value '1' in Position 10 of the P-type user input line.

5. List of entry points by impacted search criterion for each impacted field: list of entry points and impact search criteria which originated the impact, after each iteration.

>>> Report requested by value '1' in Position 14 of the P-type user input line.

6. Statistics:

   Number of impacted lines sorted by library and by entity type, all lines considered.

   >>> Report requested by value '1' in Position 11 of the P-type user input line.

7. Character-string analysis:

   List of uses of each of the character strings searched by the ISOS procedure.

   >>> Report requested by value '1' in Position 19 of the P-type user input line.

8. Operator analysis:

   List of uses of each of the operators searched by the ISOS procedure.

   >>> Report requested by value '1' in Position 20 of the P-type user input line.

9. List of entities impacted by entry point:

   List of entities impacted by Data-Element type entry points, all search criteria considered.

   >>> Report requested by value '1' in Position 21 of the P-type user input line.

10. Number of modified lines, dispatched by Description for each entity:

    This summary report allows for finer statistics by line types, compounded by library.

    >>> Report requested by value '1' in Position 22 of the P-type user input line.

11. Constant analysis:

    List of uses of each constant searched by the ISOS procedure.

    >>> Report requested by value '1' in Position 23 of the P-type user input line.

EXECUTION CONDITIONS

None, but the FO file must exist and must not be empty.

ABNORMAL EXECUTION

Whatever the cause of the abend, you can run the procedure as it is, after the problem has been solved.

7.6.2. IPIA: USER INPUT

IPIA: USER INPUT

A line identifying the context (* line) is required. It must be inserted at the beginning of the generated stream.

If you specified a lowest library for the ISEP procedure, it must be repeated in this line.

The *-type line must be followed by one P-type, formatted as follows:

Pos.	Len.	Value	Meaning
2	1	'P'	Line code
3	1		NOTHING TO ENTER,EXCEPT FOR DOS/VSE
		'I'	Default option for all hardware
		'N'	If CURRENT-DATE = DD/MM/YY
4	3	bbb	Library code (this selection is available with requests entered in Positions 9 and 10 only)
7	1	' ' '1'	Result of impact analysis by entry point
8	1	' ' '1'	List of impacted criteria by entry point
9	1	' ' '1'	Printing of results formatted as batch update transactions, sorted per Library
		'2'	Same list with page and line skips
10	1	' ' '1'	Summary of impacted occurrences
11	1	' ' '1'	Statistics, sorted per Library
12	1	' ' '1'	Identical to '1' in Position 9 but output is a file instead of print
13	1	' ' '1'	General option: Inhibits the lines indirectly impacted (e.g. -CD)
14	1	' ' '1'	List of entry points by impact search criterion
15	2	nn	Number of the wanted level (IANA iteration)
17	2	pp	Number of lines printed per page
19	1	' ' '1'	Result of character-string analysis
20	1	' ' '1'	Result of operator analysis
21	1	' ' '1'	Impacted entities by entry point
22	1	' ' '1'	Number of lines per description
23	1	' ' '1'	Constant-analysis result

USER INPUT (CONTINUED)

Pos.	Len.	Value	Meaning
24	1	' ' '1'	Result of group fields
25	10		Selection of generated transactions
		Blank	Selection of all entities
		other	Requested selection, where possible values (compoundable) are:
		'B'	Database blocks
		'E'	Data-Elements
		'F'	User Entities
		'O'	Screens, C/S screens,..
		'P'	Programs
		'R'	Reports
		'S'	Segments and Data-Structures
		'T'	Texts
		'V'	Volumes
		'$'	User Entity Occurrences
35	1	' ' '1'	Result with ISOS transactions

7.6.3. IPIA: DESCRIPTION OF STEPS

IPIA: DESCRIPTION OF STEPS

```
PRINTING OF IMPACT RESULTS:  PAN270

.Permanent input files:
 -Error messages
  PAC7AE
 -Impact results
  PAC7FO

.Transaction file:
 -User input
  PAC7MB

.Output file:
 -Generated batch transactions
  PAC7MV

.Output report:
 -Analysis results
  PAC7IF

.Sort file(s):
 Not assigned

.Return codes:
 - 0:  OK
 -12:  System error



PRINTING OF GENERATED TRANSACTIONS:  PAN280

.Permanent input files:
 -Error messages
  PAC7AE

.Transaction file:
 -User input
  PAC7MB
 -Generated batch transactions
  PAC7MV

.Output files:
 -Selected batch transactions
  PAC7VM

.Output report
 -List of transactions by library
  PAC7IT

.Return codes:
 - 0:  OK
 -12:  System error
```

7.6.4. IPIA: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) IPIA BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        IPIA 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 : Pac/Impact - PRINTING RESULTS
# **************************************************************
# * INPUT TRANSACTION FORMAT :
# * .* LINE WITH USER CODE AND PASSWORD (MANDATORY)
# *
# * .EDIT COMMAND LINE (MANDATORY)
# * COL  2     : 'P' LINE CODE
# * COL  3     :     NOTHING TO ENTER EXCEPT FOR DOS/VSE
# *            : 'I' DEFAULT OPTION FOR ALL HARDWARE
# *            : 'N' IF CURRENT-DATE = DD/MM/YY
# * COL  4-6   : 'BBB' LIBRARY CODE (THIS SELECTION AVAILABLE
# *            : WITH REQUESTS ENTERED IN POSITIONS 9 AND
# *            : AND 10 ONLY)
# * COL  7     : ' '
# *            : '1' IMPACT ANALYSIS RESULTS BY ENTRY POINT
# *            :
# * COL  8     : ' '
# *            : '1' LIST OF IMPACTED CRITERIA BY ENTRY POINT
# *            :
# * COL  9     : ' '
# *            : '1' PRINTING OF RESULTS FORMATTED AS
# *            :     BATCH UPDATE TRANSACTIONS, SORTED
# *            :     PER LIBRARY
# *            : '2' PRINTING WITH PAGE AND LINE SKIPS
# * COL  10    : ' '
# *            : '1' SUMMARY OF IMPACTED OCCURRENCES
# * COL  11    : ' '
# *            : '1' STATISTICS, SORTED PER LIBRARY
# * COL  12    : ' '
# *            : '1' GENERATING BATCH MOVEMENTS
# * COL  13    : ' '
# *            : '1' INHIBITS THE LINES INDIRECTLY  (-CD..)
# *            :
```

```
# * COL  14    : ' '
# *            : '1' LIST OF ENTRY POINTS BY IMPACT
# *            :     SEARCH CRITERION
# ***************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7FO.ini
PAC7MB=$PACINPUT'MBIPIA'
export PAC7MB
PAC7MV=$PACINPUT'MVIPIA'
export PAC7MV
PAC7IF=$PACTMP'IPIAIF.270'
export PAC7IF
echo "Execution : PAN270"
cobrun PAN270
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN270"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN270"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBIPIA'
export PAC7MB
PAC7VM=$PACINPUT'VMIPIA'
export PAC7VM
PAC7MV=$PACINPUT'MVIPIA'
export PAC7MV
PAC7IT=$PACTMP'IPIAIT.280'
export PAC7IT
echo "Execution : PAN280"
cobrun PAN280
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN280"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN280"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
echo "End of procedure"
exit $RETURN
```

7.7. IGRA: BREAKING DOWN OF GROUP FIELDS

7.7.1. IGRA: INTRODUCTION

IGRA - OVERVIEW

The IGRA procedure breaks down group fields into Elementary Fields:

1. Entry points detected by the ISEP procedure, if they are of the Group type.
2. Impact search criteria obtained by running the IANA procedure, if they are of the Group type.

The IGRA procedure is optional and does not generate any impact search criterion.

Before running the IGRA procedure, you may purge:

1. Entry points --after execution of the ISEP procedure.
2. Impact search criteria --after execution of the IANA procedure.

In both cases, deletions are made in the FR file under an editor) by inhibiting them (value 'E' in the action code of the corresponding lines), in order to save them for future executions of IANA.

It is not necessary to eliminate non-Group fields since they will simply be ignored by the procedure.

The notions of 'level' and 'iterations' are not relevant for the IGRA procedure.

Entry points (first iteration) or impact search criteria (further iterations) are printed once the purged criteria have been taken into account. This printout sorts criteria into 'accepted' and 'rejected' criteria'.

The impact results file may either be empty or contain the results of other IANA, ISOS, or IGRA executions, either in the same execution context or in different contexts. This allows you to compound the results of all iterations of the impact analysis for one or several contexts.

Restitution of all the information for a given context may be customized (parameter setting) when printing with the IPIA procedure.

The file of Entities to be analyzed (FP) is used as input to this procedure. It contains a list of Entities or Entity Types which should be analyzed. If no user input is entered in this file before its initialization by the INFP procedure, all analyzable Entities will be analyzed.

Entities to be analyzed are specified as follows: 3-character Type, and 6-character code (****** being the Entity generic code).

EXECUTION CONDITIONS

None, except that the FH file (entry points or impact search criteria) must exist and must not be empty.

ABNORMAL EXECUTION

Whatever the reason for the abnormal ending, the procedure may be resumed as it is after correcting the problem. However, the status of generation files (FH, FR, and FO) should be checked.

USER INPUT

The IGRA procedure requires no specific user input for its execution.

7.7.2. IGRA: DESCRIPTION OF STEPS

IGRA: DESCRIPTION OF STEPS

```
RECOGNITION OF PURGED CRITERIA:  PAN230

.Permanent input files:
 -Search criteria file
  PAC7FH
 -Reduced file of purged criteria
  PAC7FR

.Output file:
 -Search criteria file
  PAC7HF


PRINTING ENTRY POINTS:  PAN220

.Permanent input files:
 -Error messages
  PAC7AE
 -Sorted criteria
  PAC7HF

.Output reports:
 -List of accepted/rejected criteria
  PAC7IL

.Sort files:
  Not assigned


GROUP FIELD BREAKING-DOWN:  PAN255

.Permanent input files:
 -Error messages
  PAC7AE
 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Entities to be analyzed
  PAC7FP

.Transaction file:
 -Impacted criteria
  PAC7FH

.Output file:
 -Impact analysis results
  PAC7MF

.Return codes:

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

UPDATE OF IMPACT ANALYSIS RESULTS: PAN260

.Transaction file:
 -Impact analysis result (by level)
  PAC7MF

.Permanent input file:
 -Results of previous analysis
  PAC7OF

.Permanent output file:
 -Sorted results of the impact analysis
  PAC7FO

.Sort files:
  Not assigned

.Return codes:

 . 0 : OK.
 . 12 : System error

7.7.3. IGRA: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--         Release xxx Version xxx         --
#@(#)
#@(#)VA Pac (R) IGRA BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        IGRA 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 : Pac/Impact - SUB-DESCRIPTION OF DATA GROUPE
# **************************************************************
# * NO USER INPUT
# **************************************************************
. $PACDIR/assign/$1/PAC7FH.ini
. $PACDIR/assign/$1/PAC7FR.ini
PAC7HF=$PACTMP'HF'
export PAC7HF
echo "Execution : PAN230"
cobrun PAN230
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN230"
 echo "ERREUR 12 : System Error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN230"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PAC7HF=$PACTMP'HF'
export PAC7HF
PAC7IL=$PACTMP'IGRAIL.220'
export PAC7IL
```

```
echo "Execution : PAN220"
cobrun PAN220
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN220"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN220"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7FP.ini
PAC7HF=$PACTMP'FH'
export PAC7HF
PAC7MF=$PACTMP'MF'
export PAC7MF
PAC7FH=$PACTMP'HF'
export PAC7FH
echo "Execution : PAN255"
cobrun PAN255
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN255"
 echo "ERREUR 12 : System Error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN255"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7FO.ini
PAC7MF=$PACTMP'MF'
export PAC7MF
PAC7OF=$PAC7FO
export PAC7OF
PAC7FO=$PAC7FO.NEW
export PAC7FO
echo "Execution : PAN260"
cobrun PAN260
RETURN=$?
case $RETURN in
0)
 ;;
*)
```

```
  echo "Error in executing PAN260"
  echo "Error $RETURN"
  sh $PACDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
  ;;
 esac
 # **************************************************************
 echo "End of procedure"
 echo ""
 echo "Call of FOBACKUP.ini file"
 . $PACDIR/assign/$1/FOBACKUP.ini
 echo ""
 echo "Deletion of the temporary files"
 rm -f  $3+MF'
 rm -f  $3+HF'
 rm -f  $3+FH'
 exit $RETURN
```

7.8. IPFQ: FQ FILE PRINTOUT (IMPACT ANALYSIS)

7.8.1. IPFQ: INTRODUCTION

IPFQ: INTRODUCTION

The IPFQ procedure prints all the entry points and impact search criteria used (accepted or rejected) during a thorough impact analysis.

All the criteria and entry points are stored in the FQ file.

IPFQ offers four types of printouts:

. List of accepted entry points
. List of rejected entry points
. List of accepted impact search criteria
. List of rejected impact search criteria.

The printout shows criteria and entry points sorted by alphabetical order within each category, and by definition library of the criteria.

The printing order for the categories are:

. Character strings
. Data-Element defined in the Dictionary,
. Data-Element defined in Segment Descriptions,
. Data-Element defined in Report Structures,
. Data-Element defined in Screen- or Program-Working sections.

The IPFQ procedure can be used to select the entry points and impact search criteria of one or more categories.
In case of selection, only the selected criteria are printed.

EXECUTION CONDITIONS

None, but the FQ file must exist.

ABNORMAL EXECUTION

Whatever the cause of the abnormal ending, the procedure may be re-run as it is, after correction of the problem.

7.8.2. IPFQ: USER INPUT

IPFQ: USER INPUT

One 'S' line per criteria selection (optional):

Pos.	Len.	Value	Meaning
2	1	'S'	Line code
3	1		Type of criterion
		'E'	Data-Element defined in the Dictionary
		'C'	Character string
		'X'	Group-type Data-Element or Data-Element not defined
		'*'	All types of criteria
4	1		Source code
		'3'	Line from Segment's -CE
		'6'	Line from Report's -CE
		'7'	-W line of a Screen or Program
		'*'	All sources
6	1		For the type of field
		'G'	For a Group field
		' '	For an elementary field
		'*'	For all types of fields

7.8.3. IPFQ: DESCRIPTION OF STEPS

IPFQ: DESCRIPTION OF STEPS

```
EXTRACTION OF CRITERIA:  PAN240

.Permanent input files:
 -Error messages


 -Data file
  PAC7AR
 -Index file
  PAC7AN
 -Criteria impacted during analysis
  PAC7FQ

.Transaction file:
 -Input

.Output files:
 -Search criteria
  PAC7FH

.Output report:
 -Control report


PRINTING OF IMPACTED CRITERIA:  PAN220

.Permanent input files:
 -Error messages
  PAC7AE
 -Sorted entry points or criteria
  PAC7HF

.Output report:
 -List of entry points or criteria
  PAC7IL

.Sort file(s):
  Not assigned

.Return codes:
 - 0:  OK
 -12:  System error
```

7.8.4. IPFQ: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) IPFQ BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        IPFQ 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 : Pac/Impact - PRINTING OF THE FQ FILE
# **************************************************************
. $PACDIR/assign/$1/PAC7FQ.ini
. $PACDIR/assign/$1/PAC7FH.ini
. $PACDIR/assign/$1/PAC7AN.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBIPFQ'
export PAC7MB
PAC7IX=$PACTMP'IPFQIX.240'
export PAC7IX
echo "Execution : PAN240"
cobrun PAN240
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN240"
 echo "Error 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN240"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7FH.ini
```

```
PAC7HF=$PAC7FH
export PAC7HF
PAC7IL=$PACTMP'IPFQIL.220'
export PAC7IL
echo "Execution : PAN220"
cobrun PAN220
RETURN=$?
case $RETURN in
0)
 ;;
12)
 echo "Error in executing PAN220"
 echo "ERREUR 12 : System error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
*)
 echo "Error in executing PAN220"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
echo "End of procedure"
exit $RETURN
```

7.9. INFQ: FQ FILE REINITIALIZATION (IMPACT ANALYSIS)

7.9.1. INFQ: INTRODUCTION

INFQ: INTRODUCTION

The INFQ procedure reinitializes the FQ file, which accumulates all the search criteria that have already been impacted by the analysis. This accumulation prevents these criteria from being analyzed again in future analyses.

This action should be performed before a new impact analysis either because the entry points have changed or because the analysis context has changed.

However, it must not be used between two iterations of the same impact analysis.

7.9.2. INFQ: DESCRIPTION OF STEPS

INFQ: DESCRIPTION OF STEPS

```
REINITIALIZATION OF THE FQ FILE:  PAN200

.Output file:
 -Reinitialized impactd criteria file (sequential)
  PAC7FQ
```

7.9.3. INFQ: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) INFQ BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        INFQ PROCEDURE"
echo "                        =============="
. $PACDIR/assign/$1/PAC7FQ.ini
echo "Directory of file FQ              : $PAC7FQ"
echo "---------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/PAUSE.ini
echo ""
# **************************************************************
# * VA Pac : Pac/Impact - INITIALIZATION FQ FILE
# **************************************************************
# * NO USER INPUT
# **************************************************************
echo "Execution : PAN200"
cobrun PAN200
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PAN200"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
exit $RETURN
```

7.10. INFP: FP FILE INITIALIZATION (IMPACT ANALYSIS)

7.10.1. INFP: INTRODUCTION

INFP: INTRODUCTION

The INFP procedure initializes the FP file. It allows to specify the entities which are to be analyzed and thus to narrow the scope of the impact analysis to some (or all) occurrences of the entities.

For the FP file to be updated by INFP, you must re-state in the procedure's input all the lines previously introduced. You always start with an empty file, i.e. a file containing no particular selection.

Operating principles of the FP file's input:

If an entity type is specified (whether its specific occurrences are specified or not), and if you wish the analysis to take into account other types as well, you must explicitly specify those types (there again, with the ****** generic code if all entities of a type are required, or specific entity codes for a narrower selection).

If an entity type is coded for all its occurrences -- with the ***** code-- you cannot specify a particular entity of this type.

7.10.2. INFP: USER INPUT

INFP: USER INPUT

Input is optional for the INFP procedure knowing that if no input is provided, all entities of all entity types will be searched for the impact analysis.

If all existing entities of a given entity type are specified (code = ******), particular entities specified for the same type will be refused.

Pos.	Len.	Value	Meaning
1	3		Entity type Possible values are:
		'B '	Database Blocks
		'F '	User Entities
		'O '	Screens
		'P '	Programs
		'T '	Texts
		'V '	Volumes
		'$nn'	User Entity Occurrence of type code 'nn'
		'$**'	All UEOs
4	6		Entity code (generic selection through code ******) (This code may not exist in the Database)

7.10.3. INFP: DESCRIPTION OF STEPS

INFP: DESCRIPTION OF STEPS

```
CHECK ON TRANSACTIONS AND FP UPDATE:  PAN205

.Permanent input file:
 -Error messages
  PAC7AE

.Transaction file:
 -User input
  PAC7MB

.Output file:
 -Entities in production
  PAC7FP

.Output report:
 -Check report
  PAC7IP

.Sort file(s):
  Not assigned


 - 0:  OK
 -12:  System error
```

7.10.4. INFP: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) INFP BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        INFP PROCEDURE"
echo "                        =============="
. $PACDIR/assign/$1/PAC7FP.ini
echo "Directory of file FP              : $PAC7FP"
. $PACDIR/assign/$1/PACINPUT.ini
echo "Directory 'input'                 : `dirname $PACINPUT.`"
echo "---------------------------------------------------------"
echo ""
sh $PACDIR/batch/proc/PAUSE.ini
echo ""
# **************************************************************
# * VA Pac : Pac/Impact - INITIALIZATION FP FILE
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
PAC7MB=$PACINPUT'MBINFP'
export PAC7MB
PAC7IP=$PACTMP'INFPIP.205'
export PAC7IP
echo "Execution : PAN205"
cobrun PAN205
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PAN205"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
exit $RETURN
```

8. VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE

8.1. VDWN: RESTORATION

8.1.1. VDWN: INTRODUCTION

VDWN: INTRODUCTION

This procedure restores the VisualAge Smalltalk objects whose sources, produced by the VisualAge Smalltalk Export function, have been previously backed up in VisualAge Pacbase.

The procedure produces two files:

1. The restoration file of the objects extracted from VisualAge Pacbase. This file must be transferred onto the VisualAge Smalltalk WorkStation. It is then processed again by the local restoration procedure step, to produce a source file which will be recognized by the VisualAge Smalltalk Import function.

2. The command file for the generation of the Logical View Proxys used in the extracted objects. It can be used to re-generate the Logical View Proxys if needed.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Refer to chapter 'OVERVIEW', subchapter 'Abnormal Endings' in the 'Batch procedures Manual: the Administrator's Guide'.

8.1.2. VDWN: USER INPUT

VDWN: USER INPUT

1. Line defining the VisualAge Pacbase library-session to be processed.

Pos.	Len.	Value	Meaning	(*)
2	1	'*'	Line code	R
3	8		User code	R
11	8		Password	R
19	3		VA Pac library code	R
22	5		Session number and status	O
		SPACE	Current session	

(*) R = Required, O = Optional

2. Extraction command line (one line per object)

Pos.	Len.	Value	Meaning	(*)
2	2	'Y3'	Line code	R
4	2		Object's class	R
		'77'	VisualAge Smltlk. application	
		'78'	Part	
6	6		VA Pac identifier of the VisualAge Smalltalk object	R

8.1.3. VDWN: DESCRIPTION OF STEPS

VDWN: DESCRIPTION OF STEPS

```
CHECK AND EXTRACTION PREPARATION:  PVA100

.Input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error messages
  PAC7AE
 -User input
  PAC7MB

.Output reports and files:
 -Check report
  PAC7ET
 -'*'-line check report
  PAC7DD
 -Proxy-generation requests (GPRT)
   (Length= 80)
```

This file is used to store the requests for the generation of Logical View Proxies, Folder View Proxies, and Elementary Proxies in case these Proxies are used in the objects to be extracted. These requests can be used as input to the GPRT procedure.

```
  PAC7ME
 -Elementary-extraction requests
  PAC7MV

EXTRACTION:  PVA110

.Input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error messages
  PAC7AE
 -Elementary extraction requests
  PAC7MV

.Output file:
 -Result of host restoration
   (Length= 100)
```

This file stores the unformatted sources of extracted objects. It should be transferred onto the local workstation, in order to terminate the process with the local restoration step which is performed in the same environment as VisualAge Smalltalk.

```
  PAC7MX
```

8.1.4. VDWN: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) VDWN BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        VDWN 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 : VISUAL BRIDGE - DOWNLOAD
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7MB=$PACINPUT'MBVDWN'
export PAC7MB
PAC7ME=$PACINPUT'MEVDWN'
export PAC7ME
PAC7MV=$PACTMP'MV'
export PAC7MV
PAC7DD=$PACTMP'VDWNDD.100'
export PAC7DD
PAC7ET=$PACTMP'VDWNET.100'
export PAC7ET
echo "Execution : PVA100"
cobrun PVA100
RETURN=$?
case $RETURN in
0)
 ;;
10)
 echo "Error in executing PVA100"
 echo "All the '*' lines are rejected"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
8)
 echo "Error in executing PVA100"
 echo "At least one '*' line with severe error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
4)
```

```
 echo "Error in executing PVA100"
 echo "At least one '*' line with warning"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
*)
 echo "Error in executing PVA100"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7MV=$PACTMP'MV'
export PAC7MV
PAC7MX=$PACINPUT'MVVDWN'
export PAC7MX
echo "Execution : PVA110"
cobrun PVA110
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PVA110"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
echo ""
echo "The output file MVVDWN will be processed by VISUAL"
echo "utilities (created in the directory $PACINPUT)"
echo "MEVDWN file contains Printing-Generation commandes for"
echo "proxy (created in the directory $PACINPUT)"
echo ""
echo "Deletion of the temporary files"
rm -f  $PACTMP'MV'
exit $RETURN
```

8.2. VUP1: BACKUP - CODE CALCULATION

8.2.1. VUP1: INTRODUCTION

VUP1: INTRODUCTION

This procedure creates the elements which will be used in input to the VUP2 procedure to generate the backup transactions in VisualAge Pacbase. These transactions will be used in input to the UPDT procedure.

The VUP1 procedure creates three files:

1. a correspondence file: correspondences between the VisualAge Pacbase codes and the VisualAge Smalltalk/Java identifiers for the entities already backed up in VisualAge Pacbase.

2. New-code file: contains the VisualAge Pacbase codes computed for the new VisualAge Smalltalk/Java entities to be created during the processing with their identifiers. These computed codes may be modified if they do not meet the site's standards.

3. Transaction file: similar to the file resulting from the local backup procedure step, but with the duplicates removed.

It prints 3 reports:

1. One report showing the correspondences between VisualAge Pacbase and VisualAge Smalltalk/Java codes for entities already uploaded in the VisualAge Pacbase database.

2. One report showing the correspondence between VisualAge Pacbase and VisualAge Smalltalk/Java codes for entities currently being processed.

3. One check report, showing:

   - A list of entities extracted more than once by the current process,

   - Any fatal error likely to prevent the correct execution of procedures VUP1 and VUP2.

     These errors are 'contents' errors in the file provided by the 'local' system. Any error of this type suggests a problem was encountered while transferring the file from the local computer to the host. In this case, the processing is stopped.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Refer to chapter 'OVERVIEW', subchapter 'Abnormal Endings' in the 'Batch procedures Manual: the Administrator's Guide'.

8.2.2. VUP1: USER INPUT

VUP1: USER INPUT

The procedure's input file comes from the 'local' step of the backup procedure, performed in the same environment as VisualAge Smalltalk. It is a file coming from a local microcomputer.

Before executing the VUP1 procedure, you must complete the first line of this file (i.e. the 'I*' line) with:
. The user password
. The Product code and the Change number, if the VisualAge Pacbase Database is under DSMS control.

Pos.	Len.	Value	Meaning	(*)
2	2	'I*'	Line code	R
4	8		User code	R
12	8		Password	R
20	3		VA Pac library code	R
23	5		Session number and status	O
		SPACE	Current session	
58	9		Product + Change number if database under DSMS control	O

(*) R = Required, O = Optional.

CHARACTER-CORRESPONDENCE TABLE

This table is used to replace special characters in the VisualAge Smalltalk/Java identifiers with other characters --which may be stored in the Referential before calculation of the VisualAge Pacbase codes-- or, more typically, to replace a particular character with one contained in the VisualAge Smalltalk/Java identifier.

It contains as many positions as there are characters to be replaced.

Pos.	Len.	Meaning
1	1	Character to be replaced
2	1	Substitution character

Example of a table:

col 1	col 2
-	a
/	b
1	c
2	d

8.2.3. VUP1: DESCRIPTION OF STEPS

```
EXTRACTION OF VISUALAGE SMALLTALK/JAVA ENTITY CODES FROM
VISUALAGE PACBASE: PVA300

.Input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error messages
  PAC7AE
 -VisualAge Smalltalk/Java file produced by workstation
  PAC7VA

.Output reports and files:
 -Check report
  PAC7ET
 -'*'-line check report
  PAC7DD
 -Extracted codes
  PAC7VC

.Sort file(s):
 Not assigned

COMPARISON OF ENTITIES EXTRACTED FROM VA PAC
AND NEW ENTITIES TO BE CREATED IN VA PAC:  PVA305

.Input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error message file
  PAC7AE
 -VisualAge Smalltalk/Java file produced by the workstation
  PAC7VA
 -VisualAge Pacbase codes of VisualAge Smalltalk/Java
  entities already saved
  PAC7VC


  PAC7CA (CARTAB file in input database directory)

.Output reports and file:
 -List of new codes created
  PAC7ET
 -'*'-line check report
  PAC7DD
 -Printing of any fatal error and of the list of
  duplicate entity extractions
  PAC7ED
 -List of codes assigned to new VisualAge Smalltalk/Java
  entities
  PAC7VN
 -Useful VisualAge Smalltalk/Java transactions
  PAC7VG

.Sort file(s):
 Not assigned
```

CALCULATION OF VA PAC CODES FOR NEW VA SMALLTALK/JAVA
ENTITIES: PVA310

```
.Input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error message file
  PAC7AE
 -VisualAge Smalltalk/Java file produced by the workstation
  PAC7VA
 -VisualAge Pacbase codes of VisualAge Smalltalk/Java
  entities already saved
  PAC7VC


 -Character-correspondence table
  for substitution in the code calculation
  PAC7CA

.Output reports and file:
 -List of new codes created
  PAC7ET
 -'*'-line check report
  PAC7DD

.Output files:
 -List of codes assigned to new VisualAge Smalltalk/Java
  entities
  PAC7VN


 -List of VisualAge Pacbase codes of VisualAge Smalltalk/
  Java entities already saved
  PAC7VC
 -List of codes assigned to the new VisualAge Smalltalk/Java
  entities
  PAC7VV
 -File of codes assigned to entities already stored in
  VisualAge Pacbase
  PAC7VP
```

8.2.4. VUP1: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) VUP1 BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        VUP1 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 : VISUAL BRIDGE - CALCUL OF UPDATE TRANSACTIONS
# *********************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7VA=$PACINPUT'MBVUP1'
export PAC7VA
PAC7VC=$PACTMP'VC'
export PAC7VC
PAC7DD=$PACTMP'VUP1DD.300'
export PAC7DD
PAC7ET=$PACTMP'VUP1ET.300'
export PAC7ET
echo "Execution : PVA300"
cobrun PVA300
RETURN=$?
case $RETURN in
0)
 ;;
10)
 echo "Error in executing PVA300"
 echo "All the '*' lines are rejected"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
8)
 echo "Error in executing PVA300"
 echo "At least one '*' line with severe error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
4)
 echo "Error in executing PVA300"
 echo "At least one '*' line with warning"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
*)
 echo "Error in executing PVA300"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# ***************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7VA=$PACINPUT'MBVUP1'
export PAC7VA
PAC7VC=$PACTMP'VC'
export PAC7VC
PAC7VG=$PACINPUT'MBVUP2'
export PAC7VG
PAC7VN=$PACTMP'VN'
export PAC7VN
PAC7DD=$PACTMP'VUP1DD.305'
export PAC7DD
PAC7ED=$PACTMP'VUP1ED.305'
export PAC7ED
PAC7ET=$PACTMP'VUP1ET.305'
export PAC7ET
echo "Execution : PVA305"
cobrun PVA305
RETURN=$?
case $RETURN in
0)
 ;;
10)
 echo "Error in executing PVA305"
 echo "All the '*' lines are rejected"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
8)
 echo "Error in executing PVA305"
 echo "At least one '*' line with severe error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
4)
 echo "Error in executing PVA305"
 echo "At least one '*' line with warning"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
*)
 echo "Error in executing PVA305"
 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/$1/PAC7VP.ini
PAC7VA=$PACINPUT'MBVUP1'
export PAC7VA
PAC7CA=$PACINPUT'CARTAB'
```

```
export PAC7CA
PAC7VC=$PACTMP'VC'
export PAC7VC
PAC7VN=$PACTMP'VN'
export PAC7VN
PAC7VV=$PACINPUT'MCVUP2'
export PAC7VV
PAC7ET=$PACTMP'VUP1ET.310'
export PAC7ET
PAC7ED=$PACTMP'VUP1ED.310'
export PAC7ED
PAC7DD=$PACTMP'VUP1DD.310'
export PAC7DD
echo "Execution : PVA310"
cobrun PVA310
RETURN=$?
case $RETURN in
0)
 ;;
10)
 echo "Error in executing PVA310"
 echo "All the '*' lines are rejected"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
8)
 echo "Error in executing PVA310"
 echo "At least one '*' line with severe error"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
4)
 echo "Error in executing PVA310"
 echo "At least one '*' line with warning"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 ;;
*)
 echo "Error in executing PVA310"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
echo ""
echo "The output file MBVUP2 will be used as input to VUP2"
echo "(created in directory $PACINPUT)"
exit $RETURN
```

8.3. VUP2: GENERATION OF UPDT TRANSACTIONS

8.3.1. VUP2: INTRODUCTION

VUP2: INTRODUCTION

This procedure creates the VisualAge Pacbase backup transactions processed by the UPDT procedure.

It processes the 3 files produced by the VUP1 procedure, and integrates any modification made on codes by the user.

EXECUTION CONDITIONS

The VUP1 procedure must have been previously executed.

ABNORMAL EXECUTION

Refer to chapter 'OVERVIEW', subchapter 'Abnormal Endings' in the 'Batch procedures Manual: the Administrator's Guide'.

8.3.2. VUP2: USER INPUT

VUP2: USER INPUT

The VUP2 procedure includes two types of user input:

1. The USEFUL TRANSACTIONS file (output from VUP1)

This file is made up of a '*' line and lines used to generate the VisualAge Pacbase Database update transactions.

The '*' line must be completed before executing the VUP2 procedure:

. with the user password

. with the Product code and the Change number if the VisualAge Pacbase Database is under DSMS control, if this has not already been indicated in input to the VUP1 procedure.

Pos.	Len.	Value	Meaning	(*)
2	1	'*'	Line code	R
12	8		Password	R
58	9		Product + Change number if Database under DSMS control	O

(*) R = Required, O = Optional

2. The file of MODIFIED VisualAge Pacbase CODES resulting from the VUP1 procedure.

You can modify this file to assign the VisualAge Smalltalk entities a VisualAge Pacbase code different from the one automatically computed by the VUP1 procedure.

Use a text editor to perform the modification.

Pos.	Len.	Value	Meaning	(*)
55	6		New code chosen for the entity	R

(*) R = Required, O = Optional

8.3.3. VUP2: DESCRIPTION OF STEPS

VUP2: DESCRIPTION OF STEPS

```
GENERATION OF VA PAC TRANSACTIONS FOR UPDT:  PVA320

.Input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error messages
  PAC7AE
 -Useful transactions produced by VisualAge Smalltalk
  (from the workstation)
  PAC7VA
  (&VISUTIL file produced by VUP1)
 -Codes of new VisualAge Smalltalk/Java entities taken into
  account
  PAC7VN
  (&PBCOD file produced by VUP1)
 -Codes of VisualAge Smalltalk/Java entities already saved
  in VisualAge Pacbase
  PAC7VC

.Output reports:
 -List of VisualAge Pacbase codes taken into account
  PAC7ET
 -'*'-line check report
  PAC7DD


 -List of input transactions
  PAC7EM
 -List of erroneous transactions
  PAC7ER

.Output files:


 -Transactions for UPDT that include only definitions
  PAC7MY
 -Transactions for UPDT other than definitions
  PAC7MX

.Sort file(s):
 Not assigned
```

8.3.4. VUP2: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) VUP2 BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        VUP2 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 : VISUAL BRIDGE - UPLOAD
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7VA=$PACINPUT'MBVUP2'
export PAC7VA
PAC7VV=$PACINPUT'MCVUP2'
export PAC7VV
PAC7MX=$PACTMP'MX'
export PAC7MX
PAC7MY=$PACTMP'MY'
export PAC7MY
PAC7VC=$PACTMP'VC'
export PAC7VC
PAC7VN=$PACTMP'VN'
export PAC7VN
PAC7DD=$PACTMP'VUP2DD.320'
export PAC7DD
PAC7EM=$PACTMP'VUP2EM.320'
export PAC7EM
PAC7ER=$PACTMP'VUP2ER.320'
export PAC7ER
PAC7ET=$PACTMP'VUP2ET.320'
export PAC7ET
echo "Execution : PVA320"
cobrun PVA320
RETURN=$?
case $RETURN in
0)
 ;;
10)
 echo "Error in executing PVA320"
```

```
  echo "All the '*' lines are rejected"
  sh $PACDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
  ;;
 8)
  echo "Error in executing PVA320"
  echo "At least one '*' line with severe error"
  sh $PACDIR/batch/proc/ERRPAUSE.ini
  ;;
 4)
  echo "Error in executing PVA320"
  echo "At least one '*' line with warning"
  sh $PACDIR/batch/proc/ERRPAUSE.ini
  ;;
 *)
  echo "Error in executing PVA320"
  echo "Error $RETURN"
  sh $PACDIR/batch/proc/ERRPAUSE.ini
  exit $RETURN
  ;;
 esac
 # **************************************************************
 echo "End of procedure"
 echo ""
 cat $PAC7MY $PAC7MX > $PACINPUT'MVVUP2'
 echo "The output file MVVUP2 will be processed by UPDT"
 echo "(created in the directory $PACINPUT)"
 echo ""
 echo "Deletion of the temporary files"
 rm -f  $PACTMP'MX'
 rm -f  $PACTMP'MY'
 rm -f  $PACTMP'VC'*
 rm -f  $PACTMP'VN'
 exit $RETURN
```

8.4. VPUR: PURGE

8.4.1. VPUR: INTRODUCTION

VPUR: INTRODUCTION

The VPUR procedure allows the Database Manager to purge the Database from unused VisualAge Smalltalk/Java entities.

It operates in the following way: it reads the VisualAge Pacbase Database to find out VisualAge Smalltalk/Java entities that are not used, then it suggests a multiple-delete on these entities, sorted in reverse order from the VisualAge Pacbase Database order.

Entities for which deletion is suggested are the following:

1. Free Parts that do not belong to any application

2. Free Applications that do not contain any:
   -Archived Application
   -Child Application
   -Parent Application

You may specify a list of Library codes and Session numbers in order to restrict the research domain.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTION

Refer to chapter 'OVERVIEW', subchapter 'Abnormal Endings' in the 'Batch procedures Manual: the Administrator's Guide'.

8.4.2. VPUR: USER INPUT

VPUR: USER INPUT

1. User identification line (VisualAge Pacbase Manager):

Pos.	Len.	Value	Meaning	(*)
2	1	'*'	Line code	O
3	8		User code	O
11	8		Password	

2. Library- and Session- selection lines:

2.1. Selection of libraries (one line for each selected library).
If no line of this type is entered, all libraries are selected.

Pos.	Len.	Value	Meaning	(*)
2	2	'SL'	Line code	O
4	3		Code of selected library	O

2.2. Selection of Sessions (one line for each selected session).
If no line of this type is entered, all sessions are selected, including the current session.

Pos.	Len.	Value	Meaning	(*)
2	2	'SS'	Line code	O
4	5		Session code and status (current session: 9999Z)	O

(*) O = Required

8.4.3. VPUR: DESCRIPTION OF STEPS

VPUR: DESCRIPTION OF STEPS

```
GENERATION OF PURGE TRANSACTIONS:  PVA400

.Input files:
 -Index file
  PAC7AN
 -Data file
  PAC7AR
 -Error messages
  PAC7AE
 -User input
  PAC7MB

.Output reports and file:
 -List of user input
  PAC7ET
 -'*'-line check report
  PAC7DD
 -Generated purge-transactions
  PAC7MX

.Sort file(s):
  Not assigned.
```

8.4.4. VPUR: EXECUTION JCL

```
#!/bin/sh
#@(#)
#@(#)--   Release xxx Version xxx   --
#@(#)
#@(#)VA Pac (R) VPUR BATCH Procedure
#@(#)
# Parameter control
. $PACDIR/batch/proc/USAGE.ini
clear
echo ""
echo "---------------------------------------------------------"
echo "                        VPUR 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 : VISUAL BRIDGE - PURGE
# **************************************************************
. $PACDIR/assign/$1/PAC7AE.ini
. $PACDIR/assign/$1/PAC7AR.ini
. $PACDIR/assign/$1/PAC7AN.ini
PAC7MB=$PACINPUT'MBVPUR'
export PAC7MB
PAC7MX=$PACINPUT'MXVPUR'
export PAC7MX
PAC7DD=$PACTMP'VPURDD.400'
export PAC7DD
PAC7ET=$PACTMP'VPURET.400'
export PAC7ET
echo "Execution : PVA400"
cobrun PVA400
RETURN=$?
case $RETURN in
0)
 ;;
*)
 echo "Error in executing PVA400"
 echo "Error $RETURN"
 sh $PACDIR/batch/proc/ERRPAUSE.ini
 exit $RETURN
 ;;
esac
# **************************************************************
echo "End of procedure"
echo ""
exit $RETURN
```