VisualAge Pacbase 2.5

VA PAC 2.5 - BULL GCOS7/TDS
OPERATIONS MANUAL VOLUME III : USER'S GUIDE

DEPD7003251A

Note

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

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

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

First Edition (October 1998)

This edition applies to the following licensed program:

- VisualAge Pacbase Version 2.5

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

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

NOTICES

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

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

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

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

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

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

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

TRADEMARKS

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

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

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

TABLE OF CONTENTS

1. GENERAL INTRODUCTION TO THE BATCH PROCEDURES

1.1. PRESENTATION OF THE BATCH PROCEDURES USER'S GUIDE

INTRODUCTION TO THE BATCH PROCEDURE USER'S GUIDE

This manual is intended to present all the batch procedures likely to be required by the 'common' end-user.

These procedures relate more particularly to the following areas:

- Personalized extraction and automated documentation
- Quality analysis and control
- Integrity check techniques
- Pactables
- Pac/Impact
- VisualAge Smalltalk/Java - VisualAge Pacbase Interface

as well as all standard update and extraction procedures, printouts, generations, and so on.

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:

- introduction,
- execution conditions,
- abends.

. The description of the user input, processes and results obtained, as well as possible recommendation for use.

. The description of steps.

Steps related to the shift of generation files and to the formatting in library $NMLI.$LIBSU (library for temporary files output by the procedure) are implied, and therefore they are not described in this manual.

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 rights level per database (for the platforms allowing management of several user databases for a same system).

For more details, refer to the 'Batch Procedures: 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		Version of the session:
		'H'	Frozen session
		'T'	Test session
27	1		With the UPDT procedure, in case of multiple deletion:
		'N'	Print all transations including implicit transactions (Default)
		'O'	Print entered transactions and erroneous transactions
		'E'	Print erroneous transactions only

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

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

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 CONDITION

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), 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 sure that all needed space is made 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 EXECUTIONS

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 this program's files. 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

Refer to the batch forms and to the description of the input 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  :    EFN : $NMTU.$ROOT$FILEAR
 -Error message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Output report
 -Validity report (Length=079)
  PAC7DS
 -Error report (in case of errors)
  PAC7EI

.Return code:  Switch 30
  0:  The databse is valid
  1:  The database is invalid
      No other step will be executed.
```

TRANSACTION FORMATTING: PACA05

```
.Permanent input files:
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index File
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error message file
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Input transaction file:
 -Update transactions
   PAC7MB  :    EFN : TMBUPDT

.Output files:
 -Formatted transactions
   PAC7MV  :    EFN : TPAC7MV Longueur=167
  (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 : EFN : $NMTU.$ROOT$FILEAR
 -Index file
   PAC7AN : EFN : $NMTU.$ROOT$FILEAN
 -Journal file
   PAC7AJ : EFN : $NMTU.$ROOT$FILEAJ

.Permanent input files:
 -Error message file
   PAC7AE : EFN : $NMTU.$ROOT$ROOTAE
 -DSMS file of VA Pac elements
   PAC7DC : EFN : $NMTU.$ROOT$FILEDC
  (DSM variant only)

.Input transaction file:
 -Update transactions
   PAC7MV : EFN : TPAC7MV (resp. TPAC7OJ for REST RESY)

.Output report(s):
 -Update report
   PAC7IE (Length=132)
 -Erroneous-transaction list
   PAC7IF (Length=132)
 (The list of transactions belonging to a user is preceded
 by a banner specifying the user code.)

2.1.4. UPDT: EXECUTION JCL

```
COMM '*************************************************';
COMM '* BATCH UPDATE                                  *';
COMM '* ============                                  *';
COMM '*                                               *';
COMM '*    ORIGIN OF INPUT TRANSACTIONS.              *';
COMM '*    - BY DEFAULT :                             *';
COMM '*                                               *';
COMM '*    - OTHERWISE  : RESULTING FILE OF AN EXTRACTION  *';
COMM '*                   PROCEDURE STORED IN LIBRARY :    *';
COMM '*                   $NMLI.$LIBSU.               *';
COMM '*              ( 1 = CPSN, EXLI, EXPJ, EXPU, EXTR    *';
COMM '*                    EX62, RMEN, RTLO, RTUE OR TRUV) *';
COMM '* SYMBOLICS IN USE :                            *';
COMM '*     SIZEMB : USER INPUT FILE SIZE IN CYLS     (1)  *';
COMM '*     SIZEMV : THE DOUBLE OF  SIZEMB            (2)  *';
COMM '*     USER   : USER CODE                     ($USER) *';
COMM '*************************************************';
MVL  JCL,LABJCL=JCL,JLAB='&LAB'&1,
     USER='$USER',SIZEMB=1,SIZEMV=2,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
JUMP CR&&JLAB;
CRJCL:
CR   IF=*UPDT,
     OF=(TMBUPDT,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
JUMP CREND;
CR:
CR   IF=($NMLI.$LIBSU,&RFLI,SUBFILE=MBUPDT_&1&USER),
     OF=(TMBUPDT,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB),
     COMFILE=($NMLI.$LIBJCL,&RFLI,SUBFILE=PBEXPDSL),START=2;
CREND:
COMM '*** PTUBAS ***';
STEP PTUBAS,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  80;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7DS,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PACA05 ***';
STEP PACA05,FILE=($NMLI.$LIBLM,&RFLI),REPEAT;
     SZ  110;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
```

```
        ASG PAC7MB,TMBUPDT,TEMPRY,&RFTM,END=PASS;
        ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
        DEF PAC7MV,CISZ=$CISEQ,NBBUF=2,RECSZ=167,RECFORM=FB;
        ALC PAC7MV,SZ=&SIZEMV,UNIT=CYL,INCRSZ=1;
        ASG PAC7MW,TPAC7MW,TEMPRY,&RFTM,END=PASS;
        DEF PAC7MW,RECSZ=167,RECFORM=FB,CISZ=$CISEQ,NBBUF=2;
        ALC PAC7MW,SZ=&SIZEMB,UNIT=CYL,INCRSZ=1;
        ASG PAC7EI,SYS.OUT;
    ESTP;
    JUMP ERR,SW20,EQ,1;
    COMM '*** PACA15 ***';
    STEP PACA15,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
        SZ  260;
        ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AE,READLOCK=STAT;
        ASG PAC7AJ,$NMAJ.$ROOT$FILEAJ,&RFAJ,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7AJ,JOURNAL=BEFORE;
        ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
        ASG PAC7DC,$NMTU.$ROOT$FILEDC,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7DC,READLOCK=STAT;
        ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
        ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
        ASG PAC7IE,SYS.OUT;
        ASG PAC7IF,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
        ASG H_BJRNL,FILESTAT=TEMPRY,
            DVC=$DVTM,MD=$MDTM;
    ESTP;
    JUMP ERR,SW20,EQ,1;
```

2.2. UPDP: DATABASE UPDATE FROM PAF TABLES

2.2.1. UPDP: INTRODUCTION

UPDP: INTRODUCTION

The UPDP procedure performs an update of the Database 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.

ABENDS

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 PAF TABLES 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 PAF tables 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
36	3	nnn	No line numbering
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 TP 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  :    EFN : $NMTU.$ROOT$FILEAR
 -Error message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Output report
 -Validity report (Length=079)
  PAC7DS
 -Error report (in case of errors)
  PAC7EI

.Return code:  Switch 30
  0:  The databse is valid
  1:  The database is invalid
      No other step will be executed.
```

TRANSACTION FORMATTING: PAF900

```
.Permanent input files:
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index File
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error message file
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Input transaction file:
 -Update transactions
  PAC7GY  :    &EXPAF
   PAC7GY  :    &EXPAF

.Output files:
 -Formatted transactions
   PAC7MV  :    EFN : TPAC7MV Longueur=167
  (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  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Journal file
   PAC7AJ  :    EFN : $NMTU.$ROOT$FILEAJ

.Permanent input files:
 -Error message file
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -DSMS file of VA Pac elements
   PAC7DC  :    EFN : $NMTU.$ROOT$FILEDC
  (DSM variant only)

.Input transaction file:
 -Update transactions
   PAC7MV  : EFN : TPAC7MV (resp. TPAC7OJ for REST RESY)

.Output report(s):
 -Update report
   PAC7IE (Length=132)
 -Erroneous-transaction list
   PAC7IF (Length=132)
 (The list of transactions belonging to a user is preceded
 by a banner specifying the user code.)
```

2.2.4. UPDP: EXECUTION JCL

```
MVL  JCL,USER='$USER',SIZEMB=1,SIZEMV=2,
     EXPAF=FILE.EXTRACPAF,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
COMM '*** PTUBAS ***';
STEP PTUBAS,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  80;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7DS,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PAF900 ***';
STEP PAF900,FILE=($NMLI.$LIBLM,&RFLI),REPEAT;
     SZ  110;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PAC7GY,&EXPAF;
     ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,CISZ=$CISEQ,NBBUF=2,RECSZ=167,RECFORM=FB;
     ALC PAC7MV,SZ=&SIZEMV,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TPAC7MW,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MW,RECSZ=167,RECFORM=FB,CISZ=$CISEQ,NBBUF=2;
     ALC PAC7MW,SZ=&SIZEMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PACA15 ***';
STEP PACA15,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  260;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$NMAJ.$ROOT$FILEAJ,&RFAJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
     ASG PAC7DC,$NMTU.$ROOT$FILEDC,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7DC,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
```

```
        ASG PAC7IE,SYS.OUT;
        ASG PAC7IF,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
   ESTP;
   JUMP ERR,SW20,EQ,1;
```

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 : Revamping of Dialogs (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 (PACA10 and PACA20), the subchains 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.

The fifth character of the file names 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 codification 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 a suffix of LG, and is retrieved into the file with a suffix of GL. 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.)

Standard printing of volumes is retrieved from the file with a suffix of IN. The file with a suffix of GN 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 Dialog Windowing (PAF) is coded PAC7GT (record length is 180). Its name must be specified in the generation request.

EXECUTION CONDITION

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 Subchapter 'Abnormal Endings', in Chapter 'OVERVIEW' of the Batch Procedures, Administrator's Guide.

2.3.2. GPRT: STRUCTURE OF REQUESTS

GPRT: REQUEST STRUCTURE

The GPRT request consists of a three-character code.

The first character identifies the nature:

. 'L': List entities.

. 'D': Description of the entities, including the defini- tion, description and general documentation.

. 'G': Generation of source code for the entity specified (program, screen, database block, etc.).

. 'P': Print user manual or volume. The second character must be 'C', and the third 'U' or 'V'.

The second character specifies how the information is to be presented:

C- By Code.

E- To generate Error messages (used when nature = 'G').

K- By Keyword (value 'blank' in the SELECTION OF KEYWORD TYPE selects both implicit and explicit keywords; value 'L' selects implicit keywords only; value 'M' selects explicit keywords only).

N- By Name.

T- By Type.

The third character is the entity type:

```
B       Database Block
D       Data Structure
E       Data Element
F       User Entity
I       Parameterized Input Aid
K       Keyword (Thesaurus)
MC      Functional Integrity Constraint
MO      Model Object
MP      Model Property
MR      Model Relationship
O       On-line Screen
P       Program
Q       User-Defined Relationship
R       Report
S       Segment
T       Text
U       User Manual
V       Volume
$       User Entity Occurrence
```

For the WorkStation entities, the type (M) is appended with a one-letter code speciying whether it is a Property (P), an Object (O), a Relation (R) or a FIC (C).

SPECIAL REQUESTS

. FLx : Flow control card (x = entity type) (see subchapter "OPTIONAL CONTROL CARDS").

. JCL : Allows the user to set up the GPRT on-line submission JCL (See Section 'Generation/Printing Commands'.)

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

For the complete list describing all of the GPRT request commands, see Section 'Generation/Printing Commands' thereafter.

NOTE:

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

. in pre-formatted fields, with the command code,

. on a continuation line, by placing the asterisk ('*') in the CONTINUATION LINE field.

The presentation options and any possible parameters are indicated for each GPRT request command in Section 'Generation/Printing Commands' thereafter.

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 desired.

Furthermore, the print name contains a slection 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		SEQUENCE ORDER FOR PRINTING REPORTS This field is used to specify the sequence in which the requested reports selected by the user will be printed. Sub-reports of a standard report are printed in a pre-determined order which cannot be modified.
		blank	The output of description and list request commands will be printed in the sequence in which they are entered.
		A - 99	The standard reports will be sorted and printed according to this value. For specific requests for generation of entities, or printing of user manuals or volumes, the System automatically groups the following entity types together and assigns the following values for the printing order:
		90	Programs
		91	Screens
		92	Database blocks
		93	User manuals
		94	Error messages
		95	Data structures
		96	Volumes (PDM facility) NOTE: If the user attempts to modify these values, the system will ignore it without issuing an error message.
3	4		COMMAND FOR PRINT REQUEST NOTE: Input of the entity code is required or optional depending on the command. The following indicators describe the various options: (A) Required entity code input. (Batch mode col. 9) (B) Optional entity code input. If omitted, all occurrences of the entity type are listed in the user's hierarchical view. (C) Entity 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) field and pressing the ENTER key. What may be entered on this line depends on the command;

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			the user is instructed below on what options are possible. This corresponds to batch columns 31 to 80 inclusive. 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. (7) PRINT VOLUME BY CHAP/SUBCHAP AND CODE:_ __ __

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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	(B) A complete description of the text(s), including the relation of the text(s) with other texts and a list of paragraphs and their relation with other paragraphs. The information is sequenced by text code.
		L*T	List of Texts and paragraphs titles.
		DTT	(6) Description(s) of texts of the type specified. See the DCT command.
		LCT	(C) A list of texts, sequenced by text code.
		LKT	(2) A list of the texts whose names and/or explicit keywords contain the keyword(s) specified.
		LTT	(6)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			A list of texts whose type codes have been defined as specified. A list of all paragraph titles sequenced by text code.
			VOLUMES -------
		FLV	(C) (D) (4) This command is used to specify the job card and end-of-job delimiters: Flow control for volumes. Use the continuation line to define user parameters on the control cards.
		LCV	(C) List of Volumes, sequenced by code.
		LKV	(C) (2) List of Volumes selected according to the key word(s) entered on the continuation line.
		DCV	(B) Printing of the description of the Volume whose code is entered in the Entity field. When this code is not entered, the descriptions of all the Volumes are printed, sequenced by code.
		PCV	(B) (D) (7) Printing of the contents of the Volume whose code is entered in the Entity field. When this code is not entered, the contents of all the Volumes are printed, sequenced by code. For local printing in RTF format, the Volume must be generated with the C2 option. Partial printing is documented in the 'Personalized Documentation Manager' Reference Manual, Chapter 'Access Commands', subchapter 'Generation-Print'.
			USER MANUALS ------------
		DCU	(B) A complete description of user manual(s). The information is sequenced by user manual code.
		LCU	(C) A list of user manuals, sequenced by user manual code.
		LKU	(2) A list of the user manuals whose names and/or explicit keywords contain the keyword(s) specified.
		PCU	(B) (D: when the entity code has been entered)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			Print the contents of the user manual(s). To print chapter(s) only, enter the chapter code(s) on the continuation line. (NOTE: the PAGE NUMBERING OPTION value must be 'C' on the User Manual Definition screen.) 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, sequenced by Cobol name.
		LCE	(B) A list of defined elements, sequenced by element code.
		LKE	(C) (2) A list of the elements whose names and/or explicit keywords contain the keyword(s) specified.
		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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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.
			SEGMENTS --------
		LCS	(C) A list of segments sequenced by segment Code.
		LKS	(C) (2) A list of the 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 PACTABLE 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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE

		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 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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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.
			SCREENS -------
		DCO	(A) A complete description of the dialogue or screen specified including information from the dialogue complement screen, and uses of the screen in other screens. For screens, information is also provided on relevant 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) A list of the screens sequenced by screen code.
		LNO	(C) A list of the screens sequenced by type.
		LPO	(C) A list of the screens sequenced by external program name.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		LSO	(C) A list of the screens sequenced by external map name.
		LKO	(C) (2) A list of screens 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.
		GGS	(A) (D) (5) Generate a Business Component.
		GVC	(A) (D) (5) Generate a Proxy Logical View (from Business Comp.).
		FGC	(C) (D) (4) (8) This command is used to specify the job card and end-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 for business component
			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)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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 contiuation line to define user parameters on the control cards.
		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 words contain the keyword(s) specified.
		DSP	(S) Description of the selected Program produced by

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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) 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.
		PCM	(C) Description of PAF Tables for entities specific to a methodology.
			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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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 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 struc-

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			ture 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) C/S Facility: 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 containing the description of the screens to be "revamped" with Pacbase Web Connection. This command is used on a Client Dialog. NOTE: If a segment/screen suffix is entered on the continuation line of one of the four preceding commands, the error messages are generated/printed only for this segment/screen.
			JCL INTRODUCTION ---------- The JCL command can only be entered in the 'C4' screen format option.
		JCL	This indicates that the COMMAND LABEL/SYSTEM RESPONSE field will contain JCL.
			SHIFT TO UPPERCASE ----------
		UPC	This command allows for the automatic transformation

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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
		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
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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			List of possible values of methods: M for Merise D for YSM Y for Yourdon A for SSADM O for OMT F for IFW On the screen format option "4" of the GP screen, this field is displayed with the label "LINE". The JCL lines will be sorted according to the number entered in this field.
		<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. 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.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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. For some spcific generation-print requests, this field is automatically filled by VA Pac (for instance when requesting by keywords). You must then fill the print label continuation line yourself. NOTE: Up to five lines are allowed in a GPRT command: the actual command line and four continuation lines.
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 has several functions:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			- To specify keywords (see COMMAND FOR PRINT REQUEST field, note (2)). - To specify the Screen code within a Dialogue, the last character of the Report code within a Data Structure, or the Segment code within a Data Structure. - To specify the user parameters on control cards. See the USER'S Reference Manual, chapter "GENERATION AND/OR PRINTING", subchapter "ON-LINE REQUESTS".
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. 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 the type of text or database block when requesting a list or description sorted by type: LTT, DTT, LTB, DTB. 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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			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 PROGRAM GENERATION ---------- Enter the one-character code that identifies the job card to be inserted before the generated program. Default: Code entered on the Library Definition screen. NOTE: This value may be overridden on the various entity definition screens. Also see subchapter "OPTIONAL CONTROL CARDS UPDATING", chapter "DATABASE MANAGEMENT", OPTION CODE field in the USER'S Reference Manual. BATCH INPUT MODE: SHIFT TO UPPERCASE FOR USER-DEFINED DOCUMENTATION ---------- User Manuals ('U' entity) and 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 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 USER'S Reference Manual.
		$	No generation of map. (Use this value in conjunction with the CONTROL CARDS IN BACK OF MAP field.)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			BATCH INPUT MODE: SHIFT TO UPPERCASE FOR ENTITY-RELATED GPRT REQUESTS --- The output of entity-related GPRT requests is printed in uppercase characters with the UPC command.
		1	YES.
		0	NO (Default option).
15	1		CARDS IN BACK PGM/UPPERCASE SHIFT PROGRAM GENERATION --- Enter the one-character code that identifies the job card to be inserted after the generated program. Default: Code entered on the Library Definition Screen NOTE: This value may be overridden on the various entity definition screens. SHIFT TO UPPERCASE FOR ERROR MESSAGE PRINT-OUTS --- Error messages are printed in uppercase characters with the UPC command.
		1	YES.
		0	NO (Default option).
16	1		CONTROL CARDS AFTER MAP The one-character code that identifies the job card to be inserted after each generated screen map.
		$	No generation of map.
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		VOLUME SELECTION FOR PRINTING
		blank	Print the whole volume.
		C	Print the selected chapter.
		S	Print the selected subchapter.
19	2		CODE OF THE CHAPTER TO BE PRINTED Code of the chapter to be printed, or the chapter that

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			contains the subchapter to be printed.
20	2		CODE OF THE SUBCHAPTER TO BE PRINTED Code of the subchapter to be printed.
21	8		USER CODE This field is reserved for on-line use. It allows the user to initialize JCL lines for a new user. To do so, when the JCL lines are displayed, the user code displayed must be replaced with the code of the new user and the ENTER key must be pressed. Only the Database administrator (authorization level 4) is allowed to copy JCL lines.

2.3.4. GPRT: USER INPUT AND RESULTS

GPRT: INPUT-RESULTS

USER INPUT

The GPRT procedure uses 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 and print requests, refer to the corresponding subchapter above.

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 chapters.

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 the end of user documentation, which display the user code, facilitate the distribution of print-outs back to their authors.

2.3.6. GPRT: PROCESSING OF JOB STREAMS

PROCESSING THE JOB STREAMS

When there has been a generation request and the return code of the preceding step is less than 8, the generated stream must be processed to ensure the compilation, assembly and link-edit of the source code produced.

Several options are available for processing this stream. For example:

.Output of the job stream on a SAM file, ...

However, the simplest solution is to define the PUNCH$ PUN card.

The generated job stream is thus immediately submitted to the reader, without any intervention from the operator.

Other solutions are possible, the job stream processing depending on each site's standards and capacity.

Whichever solution is chosen, generated COBOL sources must be preceded and/or followed by a set of control cards predefined in the User Parameter file (AP) via the specific user parameter update transaction. (See the User's Reference Manual, Chapter "DATABASE MANAGEMENT", Subchapter 'OPTIONAL CONTROL CARDS UPDATE'.)

The PARM procedure may also be used for this purpose. (See Chapter "USER PARAMETER UPDATE", Subchapter 'CONTROL CARDS'.)

2.3.7. GPRT: EXECUTION JCL

```
MVL  $GPRT,USER='$USER',PPAF='',
     SIZEEX=05,SIZEGN=2,SIZEWK=5,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     RFTM='DVC=$DVTM,MD=$MDTM';
MVT:
CR   IF=*GPRT,
     OF=(TMBGPRT,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
MVTGP:
IV   PBUSGPR&1,($NMLI.$LIBJCL,&RFLI),$LIST,
     VL=(SIZEEX=&SIZEEX,SIZEGN=&SIZEGN,SIZEWK=&SIZEWK,
         RFTM=&RFTM,USER=&USER);
JUMP END,SW30,EQ,0;
JUMP NOB,SW21,EQ,0;
RUN  (TPAC7GB,TEMPRY,&RFTM) VL=(USER=&USER);
NOB:
JUMP NOE,SW22,EQ,0;
JUMP GE&PPAF;
GEPPAF:
STEP PAFP10,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,NBBUF=2,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAF80,TPAC7GE,TEMPRY,&RFTM;
     DEF PAF80,NBBUF=1;
     ASG COB80,TGEPPAF,TEMPRY,&RFTM,END=PASS;
     DEF COB80,NBBUF=1;
     ASG PAFREP,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
RUN  (TGEPPAF,TEMPRY,&RFTM) VL=(USER=&USER);
JUMP NOE;
GE:
RUN  (TPAC7GE,TEMPRY,&RFTM) VL=(USER=&USER);
NOE:
JUMP NOP,SW23,EQ,0;
JUMP GP&PPAF;
GPPPAF:
STEP PAFP10,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,NBBUF=2,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAF80,TPAC7GP,TEMPRY,&RFTM;
     DEF PAF80,NBBUF=1;
     ASG COB80,TGPPPAF,TEMPRY,&RFTM,END=PASS;
     DEF COB80,NBBUF=1;
     ASG PAFREP,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
```

```
    ESTP;
    RUN  (TGPPPAF,TEMPRY,&RFTM) VL=(USER=&USER);
    JUMP NOP;
    GP:
    RUN  (TPAC7GP,TEMPRY,&RFTM) VL=(USER=&USER);
    NOP:
    JUMP NOG,SW17,EQ,0;
    JUMP GG&PPAF;
    GGPPAF:
    STEP PAFP10,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
         SZ  120;
         ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
             ACC=READ,SHARE=MONITOR;
         DEF PAC7AE,NBBUF=2,READLOCK=STAT;
         ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
             ACC=READ,SHARE=MONITOR;
         DEF PAC7AN,READLOCK=STAT;
         ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
             ACC=READ,SHARE=MONITOR;
         DEF PAC7AR,READLOCK=STAT;
         ASG PAF80,TPAC7GG,TEMPRY,&RFTM;
         DEF PAF80,NBBUF=1;
         ASG COB80,TGGPPAF,TEMPRY,&RFTM,END=PASS;
         DEF COB80,NBBUF=1;
         ASG PAFREP,SYS.OUT;
         ASG PAC7EI,SYS.OUT;
    ESTP;
    RUN  (TGGPPAF,TEMPRY,&RFTM) VL=(USER=&USER);
    JUMP NOG;
    GG:
    RUN  (TPAC7GG,TEMPRY,&RFTM) VL=(USER=&USER);
    NOG:
    JUMP NOV,SW18,EQ,0;
    JUMP GV&PPAF;
    GVPPAF:
    STEP PAFP10,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
         SZ  120;
         ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
             ACC=READ,SHARE=MONITOR;
         DEF PAC7AE,NBBUF=2,READLOCK=STAT;
         ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
             ACC=READ,SHARE=MONITOR;
         DEF PAC7AN,READLOCK=STAT;
         ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
             ACC=READ,SHARE=MONITOR;
         DEF PAC7AR,READLOCK=STAT;
         ASG PAF80,TPAC7GE,TEMPRY,&RFTM;
         DEF PAF80,NBBUF=1;
         ASG COB80,TGVPPAF,TEMPRY,&RFTM,END=PASS;
         DEF COB80,NBBUF=1;
         ASG PAFREP,SYS.OUT;
         ASG PAC7EI,SYS.OUT;
    ESTP;
    RUN  (TGVPPAF,TEMPRY,&RFTM) VL=(USER=&USER);
    JUMP NOV;
    GV:
    RUN  (TPAC7GV,TEMPRY,&RFTM) VL=(USER=&USER);
    NOV:
    JUMP NOK,SW16,EQ,0;
    CR   IF=(TPAC7GK,TEMPRY,&RFTM),
         OF=($NMBU.$ROOT$FILEGK,&RFBU);
    CR   IF=($NMBU.$ROOT$FILEGK,&RFBU),
         OF=($NMBU.$ROOT$FILELK,&RFBU);
    NOK:
    JUMP NOL,SW24,EQ,0;
    CR   IF=(TPAC7GL,TEMPRY,&RFTM),
         OF=($NMBU.$ROOT$FILEGL,&RFBU);
    CR   IF=($NMBU.$ROOT$FILEGL,&RFBU),
         OF=($NMBU.$ROOT$FILELG,&RFBU);
    FILLIST IF=(TPAC7GT,TEMPRY,&RFTM,END=PASS) PRTFILE=DUMMY;
    JUMP NOGT,SEV,GE,3;
    CR   IF=(TPAC7GT,TEMPRY,&RFTM),
         OF=($NMBU.$ROOT$FILEGT,&RFBU);
```

```
    NOGT: JUMP CONTINUE;
    NOL:
    JUMP NOM,SW25,EQ,0;
    CR   IF=(TPAC7GM,TEMPRY,&RFTM),
         OF=($NMBU.$ROOT$FILEGM,&RFBU);
    JUMP CONTINUE;
    NOM:
    JUMP NON,SW26,EQ,0;
    CR   IF=(TPAC7GN,TEMPRY,&RFTM),
         OF=($NMBU.$ROOT$FILEGN,&RFBU);
    NON:
    JUMP NOD,SW27,EQ,0;
    RUN  (TPAC7GD,TEMPRY,&RFTM) VL=(USER=&USER);
    NOD:
    JUMP NOQ,SW28,EQ,0;
    RUN  (TPAC7GQ,TEMPRY,&RFTM) VL=(USER=&USER);
    NOQ:
    JUMP NOR,SW29,EQ,0;
    RUN  (TPAC7GR,TEMPRY,&RFTM);
    NOR:
    JUMP ERR,SW20,EQ,1;
```

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 TPAR  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
$$> -                        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.

2.3.9. 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  :    EFN : TMBEMLD
 -Sequential user-defined error messages
   PAC7EM       EFN : TPAC7EM
 -Data file
   PAC7AR       EFN : $NMTU.$ROOT$ROOTAR
 -VisualAge Pacbase error messages
   PAC7AE       EFN : $NMTU.$ROOT$ROOTAE

.Permanent output file:
 -User-defined error messages, indexed
   PAC7GL       EFN : $NMBU.$ROOT$FILELG

.Output reports:
 -Execution report
   PAC7IY
 -Batch-procedure authorization option
   PAC7DD
```

2.3.11. EMLD: EXECUTION JCL

```
COMM '*****************************************************';
COMM '* USER ERROR MESSAGES FILE LOADING                   *';
COMM '* ================================                  *';
COMM '*                                                    *';
COMM '* SYMBOLICS IN USE :                                 *';
COMM '*                                                    *';
COMM '*     USER   : USER CODE                    ($USER) *';
COMM '*     PAC7EM : OUTPUT ERROR MESSAGE FILE NAME        *';
COMM '*                                                    *';
COMM '*****************************************************';
MVL  PAC7EM='$NMBU.EMLD$USER',USER='$USER',SIZEEM=1,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*EMLD,
     OF=(TMBEMLD,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION FICHIER EM ***';
IV PBINALEM ($NMLI.$LIBJCL,&RFLI) VL=(&SIZEEM,&USER);
COMM '*** PACL93 ***';
STEP PACL93,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7MB,TMBEMLD,TEMPRY,&RFTM;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAC7EM,&PAC7EM,&RFBU;
     ASG PAC7GL,$NMBU.$ROOT$FILELG,&RFBU;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7IY,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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

UPDATE OF USER-DEFINED ERROR MESSAGES: PACL92

```
.Input files:
 -Sequential user-defined error messages
  PAC7EM        EFN : TPAC7EM
 -Data file
  PAC7AR        EFN : $NMBU.$ROOT$FILEAR
 -VisualAge Pacbase error messages
  PAC7AE        EFN : $NMBU.$ROOT$ROOTAE
 -Transaction file
  PAC7MB  :    EFN : TMBEMUP

.Permanent output file:
 -User-defined error message indexed file
  PAC7GL        EFN : $NMBU.$ROOT$FILELG

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

2.3.15. EMUP: EXECUTION JCL

```
COMM '*****************************************************';
COMM '* USER ERROR MESSAGE FILE UPDATE                    *';
COMM '* ============================                      *';
COMM '*                                                   *';
COMM '* SYMBOLICS IN USE :                                *';
COMM '*                                                   *';
COMM '*     USER   : USER CODE                    ($USER) *';
COMM '*     PAC7EM : OUTPUT ERROR MESSAGE FILE NAME       *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL  PAC7EM='$NMBU.EMLD$USER',USER='$USER',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*EMUP,
     OF=(TMBEMUP,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PACL92 ***';
STEP PACL92,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  160;
     ASG PAC7MB,TMBEMUP,TEMPRY,&RFTM;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAC7EM,&PAC7EM,&RFBU;
     ASG PAC7GL,$NMBU.$ROOT$FILELG,&RFBU;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7IU,SYS.OUT;
     ASG PAC7IX,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

2.3.16. PPAF: PAF PRE-PROCESSOR
2.3.16.1. PPAF: INTRODUCTION

PPAF: PAF PRE-PROCESSOR

PPAF: INTRODUCTION

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

EXECUTION CONDITION

None.

EXECUTION

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 Optional Control Cards in front/in back of generated program; the appropriate JCL must have been previously entered in the selected options, which are updated via the user parameter update transaction or 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 Preprocessor before compilation.

After the IDENTIFICATION DIVISION, each program contains a command line for the Preprocessor. This line is automatically generated by the GPRT procedure. 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  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Input file:
 -Generated programs
  PAF80   :    EFN : TPAC7GX

.Output files:
 -Generated programs to be compiled
  COB80   :    EFN : TCOB80

.Output report:
 -Execution report
  PAFREP
```

2.3.19. PPAF: EXECUTION JCL

```
MVL  PPAF,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
MVT:
CR   IF=*&1,
     OF=(TPAC7GX,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80);
STEP PAFP10,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  130;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,NBBUF=2,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAF80,TPAC7GX,TEMPRY,&RFTM,END=PASS;
     DEF PAF80,NBBUF=1;
     ASG COB80,&1,TEMPRY,&RFTM,END=PASS;
     DEF COB80,NBBUF=1;
     ASG PAFREP,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP END,SW30,EQ,0;
JUMP ERR,SW20,EQ,1;
```

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 for one of the following procedures:

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

EXECUTION CONDITION

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, with the respect of the input identification lines order. Each request is thus preceded by a '*' line, extracted transactions will be sorted in the request order).
- 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.

I M P O R T A N T:

- 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 will be 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 CONDITION

None, since the database is not directly updated.

However, if DESIGN entities are used, then locked, they must be reloaded in the database before the extraction.

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.

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 whole entities or parts of entities.

If the request is of the 'ALL' type, the entire entity will be 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 for the UPDT, UPDP or CPSN procedures (if the request is of the 'ALL', 'ONLY' or 'EXPT' type; the formatting for CPSN is not allowed). It is therefore possible to compare entities. procedures. It is therefore possible to compare entities.

EXECUTION CONDITION

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.

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' on-line field
32	4		Extraction type:
		' '	Entity alone
		'ALL '	Entity and used entities
		'ONLY'	Entity and only those 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 characters per position) containing exeptions 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 cont'd
-------------------------------------------------------------
!    !     !        !    'UEN':  User Entity               !
!    !     !        !    'URE':  User-defined Relationship !
!    !     !        !    '$tt':  User Entity Occurrence    !
!    !     !        !            ( tt = occurr. type code) !
-------------------------------------------------------------

Second line (continuation line for selections and excep-
tions):

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

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 will not be extracted since the extractor will 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 CONDITION

None.

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, application comp.,.. ('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 CONDITION

None, since the database is not directly updated.

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 CONDITION

None, since the database is not directly updated.

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 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 entities and to WorkStation entities.

Its output is a file containing update transactions, which will be used as input for the UPDT batch update procedure.

EXECUTION CONDITION

None, since the Database is not directly updated.

Batch procedure access authorization option:
Level 3 is required.

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': Texte
			'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

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).

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.

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 will not modify the program lines where this Data Element is used by this Segment, except if you specified 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. Since the MOVE has a wider impact than the RENAME, if duplicate occurrences exist within the sub-network of libraries equal or lower than the target library, these duplicates will be found and renamed by the RMEN procedure.

If this result is not convenient, it is advised to first run an RMEN/RENAME followed by a UPDT, then an 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.

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

The existence of a Program in a library whose level is higher than or equal to that of the target library is checked, as is that of called Macro-Structures, Data Structures, and 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

It is only possible to process a User Entity if no other User Entity with the same type exists in the sub-network of the target library.

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 which make up the UE Definition and Descriptions in a library higher or equal to the target library is checked, as is that of associated User Relations, if any.

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.

The existence of PacModel Objects in a library whose level is higher than or equal to that of the target library is checked, as is that of Elements/Properties called in PacModel descriptions ('-CM' and '-CE' screens).

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

The local entity type -- 3-character code -- must be entered in the ENTITY TYPE field. 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  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error-message file
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Erroneous transactions
   PAC7PJ  :    EFN : $NMBU.$ROOT$FILEPJ

.Input transaction file:
 -User input
   PAC7MB  :    EFN : TMBPQCE

.Work files:
 -User input
   PAC7BM  :    EFN : TPAC7BM
 -EXPU work file
   PAC7MM  :    EFN : TPAC7MM
 -EXPJ work file
   PAC7MJ  :    EFN : TPAC7MJ
 -RMEN work file
   PAC7TE  :    EFN : TPAC7TE
 -RMEN work file
   PAC7RE  :    EFN : TPAC7RE
 -RMEN work file
   PAC7RM  :    EFN : TPAC7RM

 -Extracted transactions
   PAC7WD  :    EFN : TPAC7WD
 -Multi-layered Extractor work file
   PAC7SY  :    EFN : &SY&USER

.Output files:
 -Extracted transactions for UPDT
   PAC7MV  :    EFN : TPAC7MV
 -Extracted transactions for REOR (EXPU)
   PAC7MR  :    EFN : TPAC7MR

 -Extracted transactions for UPDP
   PAC7GY  :    EFN : $NMBU.PACXGY
 -Extracted transactions for CPSN
   PAC7TD  :    EFN : $NMBU.EXSN&BB
 -Extracted transactions for EXUE
   PAC7UE  :    EFN : TPAC7UE

.Output reports:
 -General printout of the program stream
   PAC7IA
 -List of errors on input transactions
   PAC7DD
 -Summary reports on extractions
   PAC7ED
   PAC7EE
   PAC7EP
   PAC7EQ
   PAC7EZ

.Sort file(s):
   SWK
```

2.4.17. PACX: EXECUTION JCL

```
MVL  USER='$USER',SZWK='5',
     SIZEBI='5',BB='BB',
     SIZEGY='5',
     SY='$NMBU.SY',
     PAC7PJ='$NMBU.$ROOT$FILEPJ',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PACX,
     OF=(TMBPACX,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
IV   PBINALSY,($NMLI.$LIBJCL,&RFLI),
     VL=(SY=&SY,USER=&USER);
IV   PBINALSN,($NMLI.$LIBJCL,&RFLI),
     VL=(&SIZEBI,&BB);
IV   PBINALGY,($NMLI.$LIBJCL,&RFLI),
     VL=(&SIZEGY);
COMM '*** PACX   ***';
STEP PACX,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PJ,&PAC7PJ,&RFBU;
     ASG PAC7MB,TMBPACX,TEMPRY,&RFTM;
     DEF PAC7MB,NBBUF=1;
     ASG PAC7EU,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=&SZWK,&RFTM);
     ASG PAC7BM,TPAC7BM,TEMPRY,&RFTM;
     DEF PAC7BM,NBBUF=1;
     ASG PAC7MM,TPAC7MM,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MM,NBBUF=1;
     ASG PAC7MJ,TPAC7MJ,TEMPRY,&RFTM;
     DEF PAC7MJ,NBBUF=1;
     ASG PAC7TE,TPAC7TE,TEMPRY,&RFTM;
     DEF PAC7TE,NBBUF=1;
     ASG PAC7RE,TPAC7RE,TEMPRY,&RFTM;
     DEF PAC7RE,NBBUF=1;
     ASG PAC7RM,TPAC7RM,TEMPRY,&RFTM;
     DEF PAC7RM,NBBUF=1;
     ASG PAC7WD,TPAC7WD,TEMPRY,&RFTM;
     DEF PAC7WD,NBBUF=1;
     ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7MR,TPAC7MR,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MR,NBBUF=1;
     ASG PAC7TD,$NMBU.EXSN&BB,&RFBU;
     DEF PAC7TD,NBBUF=1;
     ASG PAC7GY,$NMBU.PACXGY,&RFBU;
     DEF PAC7GY,NBBUF=1;
     ASG PAC7UE,TPAC7UE,TEMPRY,&RFTM,END=PASS;
     DEF PAC7UE,NBBUF=1;
     ASG PAC7SY,&SY&USER,&RFTU;
     ASG PAC7IA,SYS.OUT;
```

```
         ASG PAC7DD,SYS.OUT;
         ASG PAC7EE,SYS.OUT;
         ASG PAC7EP,SYS.OUT;
         ASG PAC7EQ,SYS.OUT;
         ASG PAC7EZ,SYS.OUT;
         ASG PAC7ED,SYS.OUT;
    ESTP;
    JUMP ERR,SW20,EQ,1;
    JUMP END,SW30,EQ,1;
    LMN  SL INFILE=(TPAC7MV,TEMPRY,&RFTM),
         LIB=($NMLI.$LIBSU,&RFLI),
         COM='MV INFILE:MBUPDT_PACX'&USER',INFORM=SARF,
         TYPE=DAT,NUMBER=(1,1),REPLACE;';
    JUMP ERR,SEV,GE,3;
    LMN  SL INFILE=(TPAC7MR,TEMPRY,&RFTM),
         LIB=($NMLI.$LIBSU,&RFLI),
         COM='MV INFILE:MBREOR_PACX'&USER',INFORM=SARF,
         TYPE=DAT,NUMBER=(1,1),REPLACE;';
    JUMP ERR,SEV,GE,3;
    LMN  SL INFILE=(TPAC7UE,TEMPRY,&RFTM),
         LIB=($NMLI.$LIBSU,&RFLI),
         COM='MV INFILE:MBUPDT_EXUE'&USER',INFORM=SARF,
         TYPE=DAT,NUMBER=(1,1),REPLACE;';
    JUMP ERR,SEV,GE,3;
```

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 will 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.

ABEND

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 will be extracted are in a a parallel sub-network (UEOs managed by the WorkStation for example)

POS.	LEN.	VALUE	MEANING
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  :    EFN : $NMTU.$ROOT$ROOTAE
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR

.Input transaction file:
 -User input
   PAC7MB  :    EFN : TMBXPAF

.Permanent input file:
 -Variable skeleton-file
   PAC7SP  :    EFN : $NMBS.$ROOT$ROOTSP

.Permanent input/output file:
 -Extraction Paths
   PAC7GS  :    EFN : $NMBU.$ROOT$ROOTGS

.Output file:
 -Summary passed on to printing program
   PAC7ED  :    EFN : TPAC7ED
 -Temporary generated source
   PAC7GP  :    EFN : TPAC7GP

.Output report:
 -Execution report
   PAC7DD

.Sort file(s):
   SWK
```

EXTRACTION GENERATION: PTEX80

```
.Permenant input file:
 -Fixed skeleton file
   PAC7SF  :    EFN : $NMBS.$ROOT$ROOTSF

.Input file:
 -Source file generated by PTEX30
   PAC7GP  :    EFN : TPAC7GP

.Output file:
 -Generated source to be translated
   PAC7ST  :    EFN : TPAC7ST
```

PREPROCESSOR: PAFP10

```
.Permanent input files:
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error message file
```

```
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Input file:
 -Generated programs
  PAF80   :    EFN : TPAC7GX

.Output files:
 -Generated programs to be compiled
  COB80   :    EFN : TCOB80

.Output report:
 -Execution report
  PAFREP
```

PTEX PRINTING: PTEXD0

```
.Input files:
 -VA Pac error messages
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -PTEX30 report
   PAC7ED  :    EFN : TPAC7ED

.Permanent input/output file:
   PAC7GS  :    EFN : $NMBU.$ROOT$ROOTGS

.Output report:
 -Validation report
   PAC7RD

.Sort file(s):
   SWK
```

3.1.4. XPAF: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
MVT:
CR   IF=*XPAF,
     OF=(TMBXPAF,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80);
COMM '*** PTEX30 ***';
STEP PTEX30,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7SP,$NMBS.$ROOT$ROOTSP,&RFBS,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7SP,READLOCK=STAT;
     ASG PAC7MB,TMBXPAF,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ED,TPAC7ED,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ED,NBBUF=1;
     ASG PAC7GP,TPAC7GP,TEMPRY,&RFTM,END=PASS;
     DEF PAC7GP,NBBUF=1;
     ASG PAC7GS,$NMBU.$ROOT$ROOTGS,&RFBU;
     SWK WKDISK=(SZ=5,&RFTM);
     ASG PAC7DD,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP XED,SW30,EQ,1;
JUMP ERR,SW20,EQ,1;
COMM '*** PTEX80 ***';
STEP PTEX80,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7SF,$NMBS.$ROOT$ROOTSF,&RFBS,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7SF,READLOCK=STAT;
     ASG PAC7GP,TPAC7GP,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ST,TPAC7ST,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ST,NBBUF=1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP PAFP10,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  130;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAF80,TPAC7ST,TEMPRY,&RFTM,END=PASS;
     DEF PAF80,NBBUF=1;
```

```
        ASG COB80,TCOB80,TEMPRY,&RFTM,END=PASS;
        DEF COB80,NBBUF=1;
        ASG PAFREP,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
   ESTP;
   JUMP XED,SW30,EQ,1;
   JUMP ERR,SW20,EQ,1;
   COBOL INFILE=(TCOB80,TEMPRY,&RFTM,END=PASS),
         MAP,NSUBCK,NCKSEQ,NOBSERV,LEVEL=NSTD,
         PSEGMAX=24,DSEGMAX=24,CULIB=($NMLI.$LIBCUB,&RFLI);
   JUMP CONTINUE;
   XED:
   COMM '*** PTEXD0 ***';
   STEP PTEXD0,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
        SZ  120;
        ASG PAC7GS,$NMBU.$ROOT$ROOTGS,&RFBU;
        ASG PAC7ED,TPAC7ED,TEMPRY,&RFTM,END=PASS;
        ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AE,READLOCK=STAT;
        SWK WKDISK=(SZ=5,&RFTM);
        ASG PAC7RD,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
   ESTP;
```

3.2. XPDM: MASTER OUTLINE

3.2.1. XPDM: INTRODUCTION

XPDM: INTRODUCTION

PRINCIPLES

A Master Outline is a P-type Volume which designed to be called in another PDM Volume. Its functions are to:

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

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

If there are no errors, 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.

ABEND

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  :    EFN : $NMTU.$ROOT$ROOTAE
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR

.Input transaction file:
 -User input
   PAC7MB  :    EFN : TMBXPDM

.Permanent input/output file:
 -Extraction paths
  PAC7GS  :    EFN : $NMBU.$ROOT$ROOTGS

.Output files:
 -Report passed on to printing program
   PAC7ED  :    EFN : TPAC7ED
 -GS-update preparation
   PAC7SG  :    EFN : TPAC7SG

.Output report:
 -Execution report
   PAC7DD
```

GS UPDATE AND PRINTING OF THE MASTER OUTLINE: PTED60

```
.Input files:
 -VA Pac error messages
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Print file
   PAC7ED  :    EFN : TPAC7ED
 -GS-update preparation
   PAC7SG  :    EFN : TPAC7SG

.Permanent output file:
 -Extraction Paths
   PAC7GS  :    EFN : $NMBU.$ROOT$ROOTGS

.Output report:
 -Execution report
   ETAGP

.Sort file(s):
   SWK
```

3.2.4. XPDM: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
MVT:
CR   IF=*XPDM,
     OF=(TMBXPDM,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80);
COMM '*** PTED30 ***';
STEP PTED30,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7GS,$NMBU.$ROOT$ROOTGS,&RFBU;
     ASG PAC7MB,TMBXPDM,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ED,TPAC7ED,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ED,NBBUF=1;
     ASG PAC7SG,TPAC7SG,TEMPRY,&RFTM,END=PASS;
     DEF PAC7SG,NBBUF=1;
     SWK WKDISK=(SZ=5,&RFTM);
     ASG PAC7DD,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP END,SW30,EQ,1;
JUMP ERR,SW20,EQ,1;
COMM '*** PTED60 ***';
STEP PTED60,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7GS,$NMBU.$ROOT$ROOTGS,&RFBU;
     ASG PAC7ED,TPAC7ED,TEMPRY,&RFTM,END=PASS;
     ASG PAC7SG,TPAC7SG,TEMPRY,&RFTM,END=PASS;
     ASG ETATGP,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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 the Master Outlines and Extraction Master Paths are stored.

PREREQUISITE

To request the printing of the Master Outline and Extraction Master Path file, a 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  :    EFN : $NMTU.$ROOT$ROOTAE
 -Extraction paths
   PAC7GS  :    EFN : $NMBU.$ROOT$ROOTGS

.Input transaction file:
 -User input
   PAC7MB  :    EFN : TMBPRGS

.Output report:
 -Execution report
   PAC7DD
 -PAC7GS report
   ETATGS

.Sort file(s):
   SWK
```

3.3.4. PRGS: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
MVT:
CR   IF=*PRGS,
     OF=(TMBPRGS,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80);
COMM '*** PTEP90 ***';
STEP PTEP90,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7GS,$NMBU.$ROOT$ROOTGS,&RFBU;
     ASG PAC7MB,TMBPRGS,TEMPRY,&RFTM,END=PASS;
     SWK WKDISK=(SZ=5,&RFTM);
     ASG PAC7DD,SYS.OUT;
     ASG ETATGS,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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

Through 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.

EXECUTION CONDITION

None.

Batch procedure access authorization: Level 3 is required.

4.1.2. ACTI: 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.

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 used to define page layouts; lines and abscissas 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 DEFINITION AND VALUES

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 a statistical report.

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

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

COL(umns) ACTIVITY CALCULATION MODE

or

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

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

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

A printing character (&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 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.

&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 date (MMDDCCYY). 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=MMDDYY	YES		YES		YES	
&D2=MMDDYY	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.

When the request line entered is invalid, error messages are printed. The reader will find below the list of error mes- sages, some of which are commented.

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 WERE 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 WERE 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  :    EFN : $NMTU.$ROOT$ROOTAE
 -Journal Backup File
  PAC7PJ  :    EFN : $NMBU.$ROOT$FILEPJ

.Transaction file:
 -Update transactions
  PAC7MB  :    EFN : TMBACTI

.Output file
 -Transactions for selected reports
  PAC7ST  :    EFN : TPAC7ST

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

PRINTING OF RESULTS: PTU640

```
.Permanent input file:
 -Error Messages
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Input file:
 -Transactions for selected reports
  PAC7ST  :    EFN : TPAC7ST

.Output report:
 -Selected reports
  PAC7IV

.Sort file(s):
  SWK
```

4.1.5. ACTI: EXECUTION JCL

```
COMM '*****************************************************';
COMM '*                JOURNAL STATISTICS                 *';
COMM '*                 ==================                *';
COMM '*                                                   *';
COMM '*                                                   *';
COMM '*****************************************************';
MVL  SIZEST=2,
     PAC7PJ='$NMBU.$ROOT$FILEPJ',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*ACTI,
     OF=(TMBACTI,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP PTU630,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7MB,TMBACTI,TEMPRY,&RFTM,END=PASS;
     ASG PAC7PJ,&PAC7PJ,&RFBU;
     ASG PAC7ST,TPAC7ST,TEMPRY,&RFTM,END=PASS;
     ALC PAC7ST,SZ=&SIZEST,UNIT=CYL,INCRSZ=1;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP PTU640,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 60;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7ST,TPAC7ST,TEMPRY,&RFTM;
     ASG PAC7IV,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=&SIZEST,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
```

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  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Printing command file
  PAC7AG  :    EFN : $NMTU.$ROOT$FILEAG
 -PEI environment file ('Batch')
  PAC7AB  :    EFN : $NMTU.$ROOT$FILEAB
 -PEI environment file ('on-line')
  PAC7AC  :    EFN : $NMTU.$ROOT$FILEAC
 -Error-message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -User parameters
  PAC7AP  :    EFN : $NMBU.$ROOT$ROOTAP
 -QUALITY RULES file
  PACQMF  :    EFN : TPACQMF
 -Batch-language generation skeleton
  PAC7SC  :    EFN : $NMBS.$ROOT$ROOTSC
 -Dialog generation skeleton
  PAC7SG  :    EFN : $NMBS.$ROOT$ROOTSG
 -Map skeleton
  PAC7SS  :    EFN : $NMBS.$ROOT$ROOTSS

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

.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

  concatenated in the following files:
  TPAC7GE
  TPAC7GP
```

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

```
.Sort file(s):
  SWK
```

4.2.4. PQCA: EXECUTION JCL

```
MVL  USER='$USER',
     SIZEEX=05,SIZEWK=5,SIZEGN=05,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     PAC7FH=$NMBU.$ROOT$FILEFH;
     OF=(TPAC7ME,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
     OF=(TPACQMC,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
     SZ  160;
LMN  SL OUTFILE=(TPACQMF,TEMPRY,&RFTM,END=PASS),
        OUTDEF=(BLKSZ=4000,RECSZ=80,RECFORM=FB),
        COM='MV IL1:MIPQCE_'&USER',OUTFORM=SARF;';
         SHARE=MONITOR;
     SZ  270;
     ASG PACQIB,SYS.OUT;
     ASG PACQIE,SYS.OUT;
     ASG PACQIF,SYS.OUT;
     ASG PACQIG,SYS.OUT;
     ASG PACQMC,TPACQMC,TEMPRY,&RFTM;
     ASG PACQMF,TPACQMF,TEMPRY,&RFTM;
     ASG PACQMJ,TPACQMJ,TEMPRY,&RFTM;
     ASG PACQMK,TPACQMK,TEMPRY,&RFTM;
     ASG PACQMM,TPACQMM,TEMPRY,&RFTM;
     ASG PACQMN,TPACQMN,TEMPRY,&RFTM;
     ASG PACQMO,TPACQMO,TEMPRY,&RFTM;
     ASG PACQMZ,TPACQMZ,TEMPRY,&RFTM;
     ASG PAC7AC,$NMTU.$ROOT$FILEAC,&RFTU,
         ACC=READ,SHARE=MONITOR;
     ASG PAC7AB,$NMTU.$ROOT$FILEAB,&RFTU,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AB,NBBUF=1,JOURNAL=BEFORE;
     DEF PAC7AC,NBBUF=1,READLOCK=STAT;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,NBBUF=2,READLOCK=STAT;
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AG,NBBUF=1,JOURNAL=BEFORE,READLOCK=NORMAL;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4,READLOCK=STAT;
     ASG PAC7AP,$NMTU.$ROOT$FILEAP,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AP,READLOCK=STAT;
     ASG PAC7GG,TPAC7GG,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GG,SZ=&SIZEGN,UNIT=CYL,INCRSZ=1;
     DEF PAC7GG,CISIZE=$CISEQ,NBBUF=1;
     ALC PAC7GV,SZ=&SIZEGN,UNIT=CYL,INCRSZ=1;
     DEF PAC7GV,CISIZE=$CISEQ,NBBUF=1;
     DEF PAC7GE,CISIZE=$CISEQ,NBBUF=1;
     DEF PAC7EE,NBBUF=1;
     ASG PAC7SS,$NMBS.$ROOT$ROOTSS,&RFBS,ACC=READ;
     ASG PAC7EG,TPAC7EG,TEMPRY,&RFTM,END=PASS;
     DEF PAC7EG,NBBUF=1;
     ASG PAC7EV,TPAC7EV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7EV,NBBUF=1;
     ASG PAC7EP,TPAC7EP,TEMPRY,&RFTM,END=PASS;
     DEF PAC7EP,NBBUF=1;
     ASG PAC7GP,TPAC7GP,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GP,SZ=&SIZEGN,UNIT=CYL,INCRSZ=1;
     DEF PAC7GP,CISIZE=$CISEQ,NBBUF=1;
     ASG PAC7GI,TPAC7GI,TEMPRY,&RFTM,END=PASS;
```

```
        DEF PAC7GI,NBBUF=1;
        ASG PAC7SC,$NMBS.$ROOT$ROOTSC,&RFBS,ACC=READ;
        ASG PAC7JG,TPAC7JG,TEMPRY,&RFTM;
        ALC PAC7JG,SZ=04,UNIT=TRACK,INCRSZ=04;
        DEF PAC7JG,NBBUF=1;
        ASG PAC7KG,TPAC7KG,TEMPRY,&RFTM;
        ALC PAC7KG,SZ=&SIZEEX,UNIT=CYL,INCRSZ=1;
        DEF PAC7KG,NBBUF=1;
        ASG PAC7KV,TPAC7KV,TEMPRY,&RFTM;
        ALC PAC7KV,SZ=&SIZEEX,UNIT=CYL,INCRSZ=1;
        DEF PAC7KV,NBBUF=1;
        ASG PAC7KD,TPAC7KD,TEMPRY,&RFTM;
        ALC PAC7KD,SZ=&SIZEEX,UNIT=CYL,INCRSZ=1;
        DEF PAC7KD,NBBUF=1;
        ASG PAC7KE,TPAC7KE,TEMPRY,&RFTM;
        ALC PAC7KE,SZ=&SIZEEX,UNIT=CYL,INCRSZ=1;
        DEF PAC7KE,NBBUF=1;
        ASG PAC7KF,TPAC7KF,TEMPRY,&RFTM;
        DEF PAC7KF,NBBUF=1;
        ASG PAC7KP,TPAC7KP,TEMPRY,&RFTM;
        ALC PAC7KP,SZ=&SIZEEX,UNIT=CYL,INCRSZ=1;
        DEF PAC7KP,NBBUF=1;
        ASG PAC7KS,TPAC7KS,TEMPRY,&RFTM;
        DEF PAC7KS,NBBUF=1;
        ASG PAC7KU,TPAC7KU,TEMPRY,&RFTM;
        DEF PAC7KU,NBBUF=1;
        ASG PAC7ME,TPAC7ME,TEMPRY,&RFTM;
        DEF PAC7ME,NBBUF=1;
        ASG PAC7MG,TPAC7MG,TEMPRY,&RFTM;
        DEF PAC7MG,NBBUF=1;
        ASG PAC7W1,TPAC7W1,TEMPRY,&RFTM;
        ALC PAC7W1,SZ=&SIZEWK,UNIT=CYL,INCRSZ=1;
        DEF PAC7W1,NBBUF=1;
        ASG PAC7W2,TPAC7W2,TEMPRY,&RFTM;
        ALC PAC7W2,SZ=&SIZEWK,UNIT=CYL,INCRSZ=1;
        DEF PAC7W2,NBBUF=1;
        ASG PAC7W3,TPAC7W3,TEMPRY,&RFTM;
        ALC PAC7W3,SZ=&SIZEWK,UNIT=CYL,INCRSZ=1;
        DEF PAC7W3,NBBUF=1;
        ASG PAC7W4,TPAC7W4,TEMPRY,&RFTM;
        ALC PAC7W4,SZ=&SIZEWK,UNIT=CYL,INCRSZ=1;
        DEF PAC7W4,NBBUF=1;
        ASG PAC7IH,SYS.OUT;
        ASG PAC7IA,SYS.OUT;
        ASG PAC7ID,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
        SWK WKDISK=(SZ=5,&RFTM);
    ESTP;
    JUMP ERR,SW20,EQ,1;
    JUMP END,SW30,EQ,0;
    JUMP NOE,SW22,EQ,0;
    GE:
    RUN  (TPAC7GE,TEMPRY,&RFTM) VL=(USER=&USER);
    NOE:
    JUMP NOP,SW23,EQ,0;
    GP:
    RUN  (TPAC7GP,TEMPRY,&RFTM) VL=(USER=&USER);
    NOP:
    JUMP ERR,SW20,EQ,1;
```

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 CONDITION

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 similar 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
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 libr.
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  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error-message file
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Erroneous transactions
   PAC7PJ  :    EFN : $NMBU.$ROOT$FILEPJ

.Input transaction file:
 -User input
   PAC7MB  :    EFN : TMBPQCE

.Work files:
 -User input
   PAC7BM  :    EFN : TPAC7BM
 -EXPU work file
   PAC7MM  :    EFN : TPAC7MM
 -EXPJ work file
   PAC7MJ  :    EFN : TPAC7MJ
 -RMEN work file
   PAC7TE  :    EFN : TPAC7TE
 -RMEN work file
   PAC7RE  :    EFN : TPAC7RE
 -RMEN work file
   PAC7RM  :    EFN : TPAC7RM


 -Extracted transactions
   PAC7WD  :    EFN : TPAC7WD
 -Multi-layered Extractor work file
   PAC7SY  :    EFN : &SY&USER

.Output files:
 -Extracted transactions for UPDT
   PAC7MV  :    EFN : TPAC7MV
 -Extracted transactions for REOR (EXPU)
   PAC7MR  :    EFN : TPAC7MR

 -Extracted transactions for UPDP
   PAC7GY  :    EFN : $NMBU.PACXGY
 -Extracted transactions for CPSN
   PAC7TD  :    EFN : $NMBU.EXSN&BB
 -Extracted transactions for EXUE
   PAC7UE  :    EFN : TPAC7UE

.Output reports:
 -General printout of the program stream
   PAC7IA
 -List of errors on input transactions
   PAC7DD
 -Summary reports on extractions
   PAC7ED
   PAC7EE
   PAC7EP
   PAC7EQ
   PAC7EZ
```

.Sort file(s):
SWK

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 : EFN : $NMTU.$ROOT$ROOTAE
-Data file
PAC7AR : EFN : $NMTU.$ROOT$FILEAR

.Output file:
-'Compiled' Quality Rules
PACQMI EFN : TPACQMI

.Transaction files:
-User input
PAC7MB : EFN : TMBPQCE
-User entity occurrences
PACQMC : EFN : TPAC7UE

.Output file:
-Preparation for printing
PACQML : EFN : TPACQML

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

.Sort file(s):
SWK

PRINTING OF QUALITY RULES: PTUQ15

.Permanent input file:
-Error message file
PAC7AE : EFN : $NMTU.$ROOT$ROOTAE

.Input file:
-Preparation for printing
PACQML : EFN : TPACQML

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

.Sort file(s):
SWK

4.2.8. PQCE: EXECUTION JCL

```
MVL  USER='$USER',SIZEMV='1',SZWK='3',SY='$NMBU.SY',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PQCE,
     OF=(TMBPQCE,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
IV   PBINALSY,($NMLI.$LIBJCL,&RFLI),
     VL=(SY=&SY,USER=&USER);
STEP PACX,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  60;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PJ,&PAC7PJ,&RFBU;
     ASG PAC7MB,TMBPQCE,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MB,NBBUF=1;
     ASG PAC7EU,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=&SZWK,&RFTM);
     ASG PAC7BM,TPAC7BM,TEMPRY,&RFTM;
     DEF PAC7BM,NBBUF=1;
     ASG PAC7MM,TPAC7MM,TEMPRY,&RFTM;
     DEF PAC7MM,NBBUF=1;
     ASG PAC7MJ,TPAC7MJ,TEMPRY,&RFTM;
     DEF PAC7MJ,NBBUF=1;
     ASG PAC7TE,TPAC7TE,TEMPRY,&RFTM;
     DEF PAC7TE,NBBUF=1;
     ASG PAC7RE,TPAC7RE,TEMPRY,&RFTM;
     DEF PAC7RE,NBBUF=1;
     ASG PAC7RM,TPAC7RM,TEMPRY,&RFTM;
     DEF PAC7RM,NBBUF=1;
     ASG PAC7WD,TPAC7WD,TEMPRY,&RFTM;
     DEF PAC7WD,NBBUF=1;
     ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7MR,TPAC7MR,TEMPRY,&RFTM;
     DEF PAC7MR,NBBUF=1;
     ASG PAC7TD,TPAC7TD,TEMPRY,&RFTM;
     DEF PAC7TD,NBBUF=1;
     ASG PAC7GY,TPAC7GY,TEMPRY,&RFTM;
     DEF PAC7GY,NBBUF=1;
     ASG PAC7UE,TPAC7UE,TEMPRY,&RFTM;
     DEF PAC7UE,NBBUF=1;
     ASG PAC7SY,&SY&USER,&RFTU;
     ASG PAC7IA,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7EE,SYS.OUT;
     ASG PAC7EP,SYS.OUT;
     ASG PAC7EQ,SYS.OUT;
     ASG PAC7EZ,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PTUQ10 ***';
STEP PTUQ10,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
```

```
        SZ  110;
        ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AE,READLOCK=STAT;
        ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AR,READLOCK=STAT;
        ASG PAC7MB,TMBPQCE,TEMPRY,&RFTM;
        ASG PACQMC,TPAC7MV,TEMPRY,&RFTM;
        ASG PACQMC,TPAC7UE,TEMPRY,&RFTM;
        ASG PACQMI,TPACQMI,TEMPRY,&RFTM,END=PASS;
        ASG PACQML,TPACQML,TEMPRY,&RFTM,END=PASS;
        ASG PACQIC,SYS.OUT;
        ASG PAC7DD,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
        SWK WKDISK=(SZ=&SIZEMV,&RFTM);
   ESTP;
   JUMP ERR,SW20,EQ,1;
   COMM '*** PTUQ15 ***';
   STEP PTUQ15,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
        SZ  110;
        ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AE,READLOCK=STAT;
        ASG PACQML,TPACQML,TEMPRY,&RFTM;
        ASG PACQII,SYS.OUT;
        ASG PACQIJ,SYS.OUT;
        ASG PAC7EI,SYS.OUT;
        SWK WKDISK=(SZ=&SIZEMV,&RFTM);
   ESTP;
   JUMP ERR,SW20,EQ,1;
   LMN  SL INFILE=(TPACQMI,TEMPRY,&RFTM),
        LIB=($NMLI.$LIBSU,&RFLI),
        COM='MV INFILE:MIPQCE_'&USER',INFORM=SARF,
             TYPE=DAT,NUMBER=(1,1),REPLACE;';
```

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 work station 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 CONDITION

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 DATAFLOWS 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 dataflows 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  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error-message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Transaction file:
 -User input
  PAC7MB  :    EFN : TMBSADM

.Work file(s):
 -PAF file
  SYSPAF  :    EFN : $NMBU.$ROOT$ROOTPA

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

5.1.4. SADM: EXECUTION JCL

```
MVL  SYSPAF='$NMBU.$ROOT$FILEPA',USER='$USER',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*SADM,
     OF=(TMBSADM,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
IV   PBINALPA,($NMLI.$LIBJCL,&RFLI),$LIST,
     VL=(USER=&USER,SYSPAF=&SYSPAF,TYPE='BT');
COMM '*** PADM10 ***';
STEP PADM10,FILE=($NMLI.$LIBLM,&RFLI),REPEAT;
     SZ  130;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PAC7MB,TMBSADM,TEMPRY,&RFTM,END=PASS;
     ASG SYSPAF,&SYSPAF&USER,&RFTU;
     ASG PAC7EJ,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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 Dataflow 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.

EXECUTION CONDITION

None

5.2.2. YSMC: USER INPUT

YSMC: USER INPUT

USER INPUT

One '*'-line is required for library access:

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'	"Eventflow Decomposition list"
		'8'	"Group Dataflow Decomposition list"
		'9'	"Multiple Dataflow 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)".

. "Datastore decomposition list".

. "Eventflow decomposition list".

. "Group Dataflow Decomposition list".

. "Multiple Dataflow Decomposition list".

5.2.3. YSMC: DESCRIPTION OF STEPS

YSMC: DESCRIPTION OF STEPS

YSM METHOD INTEGRITY CHECKING: PYSMCC

```
.Permanent input files:
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error-message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Transaction file:
 -User input
  PAC7MB  :    EFN : TMBPYSM

.Work file(s):
 -PAF file
  PAC7PA  :    EFN : $NMBU.$ROOT$ROOTPA&USER

.Output reports:
 -SSADM integrity checking lists
  PAC7EJ
 -SSADM & IFW validation reports
  PAC7EI
```

INTER-PROCESS CONSISTENCY: PYSMC3

```
.Permanent input files:
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error-message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Transaction file:
 -User input
  PAC7MB  :    EFN : TMBPYSM

.Work file(s):
 -PAF file
  SYSPAF  :    EFN : $NMBU.$ROOT$ROOTPA&USER

.Output report:
 -SSADM integrity-check lists
  PAC7EJ
```

LIST OF RELATIONSHIPS AND REPORTS: PYSMC2

```
.Permanent input files:
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Error messages
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Transaction file:
 -User input
  PAC7MB  :    EFN : TMBPYSM

.Work file(s):
 -PAF file
  SYSPAF  :    EFN : $NMBU.$ROOT$ROOTPA&USER
  PAC7P1  :    EFN : $NMBU.$ROOT$ROOTP1

.Output report:
 -SSADM intgrity-check lists
  PAC7EJ
```

5.2.4. YSMC: EXECUTION JCL

```
MVL  SYSPAF='$NMBU.$ROOT$FILEPA',USER='$USER',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PYSM,
     OF=(TMBPYSM,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
IV   PBINALPA,($NMLI.$LIBJCL,&RFLI),$LIST,
     VL=(USER=&USER,SYSPAF=&SYSPAF,TYPE='BT');
COMM '*** PYSMCC ***';
STEP PYSMCC,FILE=($NMLI.$LIBLM,&RFLI),REPEAT;
     SZ  130;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PAC7MB,TMBPYSM,TEMPRY,&RFTM,END=PASS;
     ASG SYSPAF,&SYSPAF&USER,&RFTU;
     ASG PAC7EI,SYS.OUT;
     ASG PAC7EJ,SYS.OUT;
     ASG PAC7EK,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PYSMC3 ***';
STEP PYSMC3,FILE=($NMLI.$LIBLM,&RFLI),REPEAT;
     SZ  130;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PAC7MB,TMBPYSM,TEMPRY,&RFTM,END=PASS;
     ASG SYSPAF,&SYSPAF&USER,&RFTU;
     ASG PAC7EJ,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PYSMC2 ***';
STEP PYSMC2,FILE=($NMLI.$LIBLM,&RFLI),REPEAT;
     SZ  130;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PAC7MB,TMBPYSM,TEMPRY,&RFTM,END=PASS;
     ASG SYSPAF,&SYSPAF&USER,&RFTU;
```

```
     ASG PAC7EJ,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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 CONDITION

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

Batch procedure 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 ')
		ttttt	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	ttttt	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 GETT procedure of the Pactables Function.

6.1.3. GETD - GETA: DESCRIPTION OF STEPS

GETD: DESCRIPTION OF STEPS

EXTRACTION & UPDATE PREPARATION: PACT40

```
.Permanent input files:
 -VisualAge Pacbase data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -VisualAge Pacbase index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -VisualAge Pacbase error-message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Table-description file
  PAC7TD  :    EFN : $NMBU.$ROOT$FILETD

.Input transaction file:
 -User requests
  PAC7MB  :    EFN : TMBGETD ou GETA

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

.Output file:
 -Descriptions update transactions higher or equal to 2.0
  PAC7MD                  $NMBU.GETA&USER
```

FORMATTING OF DESCRIPTIONS < R 2.0: PACT45

```
.Input file:
 -Description-update transactions higher or equal to 2.0
  PAC7MD                  $NMBU.GETA&USER

.Output file
 -Description-update transactions lower or equal to 2.0
  PAC7ND                  $NMBU.GETA2&USER
```

UPDATE OF TABLE-DESCRIPTION FILE: PACT50

```
(GETD procedure only)

.Permanent input file:
 -Table-description file
  PAC7TD  :    EFN : $NMBU.$ROOT$FILETD

.Input transaction files:
 -User requests
  PAC7MB  :    EFN : TMBGETD ou GETA
 -Update transactions
  PAC7MD  :    EFN : $NMBU.GETA&USER

.Output report:
 -Update review
  PAC7ET

.Sort file(s):
 SWK
```

6.1.4. GETD: EXECUTION JCL

```
COMM '*************************************************';
COMM '* TABLE DESCRIPTION GENERATION                  *';
COMM '* ============================                  *';
COMM '*                                               *';
COMM '* SYMBOLICS IN USE :                            *';
COMM '*                                               *';
COMM '*     SIZEMV : TRANSACTION FILE SIZE IN CYLS   (1)*';
COMM '*                                               *';
COMM '*************************************************';
MVL  SIZEMV=1,USER='$USER',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*GETD,
     OF=(TMBGETD,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : GT ***';
IV   PBINALTA ($NMLI.$LIBJCL,&RFLI) VL=(&SIZEMV,&USER);
COMM '*** PACT40 ***';
STEP PACT40,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7MB,TMBGETD,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7TD,$NMTU.$ROOT$FILETD,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PACT45 ***';
STEP PACT45,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7ND,$NMBU.GETA2&USER,&RFBU;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PACT50 ***';
STEP PACT50,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7TD,$NMTU.$ROOT$FILETD,&RFTU;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=2,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
```

6.1.5. GETA: EXECUTION JCL

```
COMM '*************************************************';
COMM '* TABLE DESCRIPTION EXTRACTION                  *';
COMM '* ============================                  *';
COMM '*                                               *';
COMM '* SYMBOLICS IN USE :                            *';
COMM '*                                               *';
COMM '*     USER   : USER CODE                 ($USER)*';
COMM '*     SIZEMV : TRANSACTION FILE SIZE IN CYLS      (1)*';
COMM '*************************************************';
MVL  USER='$USER',SIZEMV=1,TRTAB=$TRTAB,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*GETA,
     OF=(TMBGETA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : GT ***';
IV   PBINALTA ($NMLI.$LIBJCL,&RFLI) VL=(&SIZEMV,&USER);
COMM '*** PACT40 ***';
STEP PACT40,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  150;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7MB,TMBGETA,TEMPRY,&RFTM;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7TD,&TRTAB!!TD,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PACT45 ***';
STEP PACT45,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  150;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7ND,$NMBU.GETA2&USER,&RFBU;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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 (equivalent of GETI).

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 CONDITION

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 ENDINGS

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 modi- fication 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 used
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	ttttt	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  :    EFN : $NMTU.$ROOT$FILEAR         ***
 -VisualAge Pacbase Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN         ***
 -VisualAge Pacbase Error-message file
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE         ***
 -Table-description file
  PAC7TD  :    EFN : $NMBU.$ROOT$FILETD         ***

.Input Transaction file:
 -Descriptions requests
  PAC7MB  :    EFN : TMBGET2 ou GET1            ***

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

.Output file:
 -2.0-Description update transactions
  PAC7MD                  $NMBU.GET1&USER

.Return code(s):
```

TABLE-DESCRIPTION UPDATE: PACT51

```
(GET2 procedure only)

.Permanent input file:
 -Table-description file
  PAC7TD  :    EFN : $NMBU.$ROOT$FILEDT

.Input transaction files:
 -Descriptions requests
  PAC7MB  :    EFN : TMBGET2 ou GET1


 -Update transactions
  PAC7MD  :    EFN : $NMBU.GET1&USER

.Output report:
 -Update report
  PAC7ET

.Sort files:
 SWK
```

6.1.9. GET2: EXECUTION JCL

```
MVL  SIZEMV=1,USER='$USER',
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*GET2,
     OF=(TMBGET2,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : GT ***';
IV   PBINALTA ($NMLI.$LIBJCL,&RFLI) VL=(&SIZEMV,&USER);
COMM '*** PACT41 ***';
STEP PACT41,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  150;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7MB,TMBGET2,TEMPRY,&RFTM;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7TD,$NMTU.$ROOT$FILETD,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PACT51 ***';
STEP PACT51,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7TD,$NMTU.$ROOT$FILETD,&RFTU;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
     SWK WKDISK=(SZ=2,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
```

6.1.10. GET1: EXECUTION JCL

```
MVL  USER='$USER',SIZEMV=1,TRTAB=$TRTAB,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*GET1,
     OF=(TMBGET1,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : GT ***';
IV   PBINALTA ($NMLI.$LIBJCL,&RFLI) VL=(&SIZEMV,&USER);
COMM '*** PACT41 ***';
STEP PACT41,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  150;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7MB,TMBGET1,TEMPRY,&RFTM;
     ASG PAC7MD,$NMBU.GETA&USER,&RFBU;
     ASG PAC7TD,&TRTAB!!TD,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7TD,NBBUF=1,READLOCK=STAT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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 : EFN : $NMBU.$ROOT$FILETD

.Transaction input file:
-Parameter line
PAC7MD : TMBGETI

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
PAC7ED			
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

6.2.3. GETI: EXECUTION JCL

```
COMM '************************************************';
COMM '* TABLE DESCRIPTION INITIALIZATION               *';
COMM '* ==============================                *';
COMM '*                                                *';
COMM '************************************************';
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*GETI,
     OF=(TMBGETI,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : TD ***';
IV   PBINALTD,($NMLI.$LIBJCL,&RFLI);
COMM '*** PACTIN ***';
STEP PACTIN,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7MD,TMBGETI,TEMPRY,&RFTM;
     ASG PAC7TD,$NMTU.$ROOT$FILETD,&RFTU;
     ASG PAC7ED,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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

.Input transaction file:
-Parameter line

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

6.2.6. GET0: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*GET0,
     OF=(TMBGET0,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** ALLOCATION : TD ***';
IV   PBINALT0,($NMLI.$LIBJCL,&RFLI);
COMM '*** PACTI1 ***';
STEP PACTI1,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7MD,TMBGET0,TEMPRY,&RFTM;
     ASG PAC7TD,$NMTU.$ROOT$FILETD,&RFTU;
     ASG PAC7ED,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
```

7. PAC/IMPACT

FOREWORD

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

Impact analysis requires very large amounts 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 perfomed on this subset.

   NOTE: Extraction of a session is also possible.

   Furthermore, the REOR procedure (which must always be run after an 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 '9. 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).

SELECTION SCOPE LIMITATION

For better performance, it is advisable to limit the scope of the selection. This can be done at two levels, but in any case should be done before running the procedure:

. Via the UXSR procedure, create another VisualAge Pacbase Database. The new Database is a subset (restructured and/or renamed) of the initial Database. The analysis will be performed on this subset. (For information on the UXSR procedure, see the Administrator's Guide, Chapter 'MANAGER'S UTILITIES', Sub-chapter 'UXSR: Partial Sub-Network Extraction'.)

. Via the INFP utility --documented in Sub-Chapter 'INFP: Initialization of the FP file'-- you can restrict the selection scope to occurrences of the Program, Screen, and Database Block entities. Further selection options are also available. (For information on the INFP procedure, see Subchapter 'INFP: FP File Initialization (Impact Analysis)'.)

NOTE: If a Data Element defined in the Repository meets the ISEP selection criteria, it will always be considered by ISEP as an entry point, whatever the selection implemented via the INFP utility.

For such Data Elements, the INFP selection is effective when executing the 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 CONDITION

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  :    EFN : $NMTU.$ROOT$ROOTAE
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -File of entities to be analyzed
  PAC7FP  :    EFN : $NMBU.$ROOT$FILEFP

.Transactions file:
 -User input
  PAC7MB  :    EFN : TMBISEP

.Output file:
 -Selected entry points
  PAC7FH  :    EFN : TPAC7FH  Longueur=160

.Output report(s):
 -Validation report
  PAC7IE
```

REMOVAL OF DUPLICATE ENTRY POINTS: PAN215

```
.Transactions file:
 -Selected entry points
  PAC7FH  :    EFN : TPAC7FH

.Permanent output files:
 -Sorted selected entry points
  PAC7HF  :    EFN : $NMBU.$ROOT$FILEFH/G+1
 -Reduced entry points to be purged
  PAC7FR  :    EFN : $NMBU.$ROOT$FILEFR/G+1

.Sort file(s):
  SWK
```

7.1.4. ISEP: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM',
     SIZEWK=2,
     PAC7FH=$NMBU.$ROOT$FILEFH,
     PAC7FR=$NMBU.$ROOT$FILEFR;
CR   IF=*ISEP,
     OF=(TMBISEP,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PAN210 ***';
STEP PAN210,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAC7FP,$NMBU.$ROOT$FILEFP,&RFBU,
         SHARE=MONITOR;
     DEF PAC7FP,READLOCK=STAT;
     ASG PAC7MB,TMBISEP,TEMPRY,&RFTM;
     ASG PAC7FH,TPAC7FH,TEMPRY,&RFTM,END=PASS;
     ALC PAC7FH,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PAN215 ***';
STEP PAN215,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 110;
     ASG PAC7FH,TPAC7FH,TEMPRY,&RFTM;
     ASG PAC7HF,&PAC7FH/G+1;
     ASG PAC7FR,&PAC7FR/G+1;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** SHIFT FILES ***';
SHIFT &PAC7FH;
SHIFT &PAC7FR;
```

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 CONDITION

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  :    EFN : $NMTU.$ROOT$ROOTAE
 -Entry points
  PAC7HF  :    EFN : $NMBU.$ROOT$FILEFH

.Output report:
 -List of entry points
  PAC7IL

.Sort file(s):
  SWK
```

7.2.3. IPEP: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     RFTM='DVC=$DVTM,MD=$MDTM',
     PAC7FH=$NMBU.$ROOT$FILEFH;
COMM '*** PAN220 ***';
STEP PAN220,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7HF,&PAC7FH;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IL,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
```

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).

LIMITATION OF SELECTION DOMAIN

For better performance, it is advisable to limit 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, by creating 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. (For information on the UXSR procedure, see the Administrator's Guide, Chapter 'MANAGER'S UTILITIES', Sub-chapter 'UXSR: Partial Sub-Network Extraction'.) "INFP : Initializing FP file (procedure PBININIT)",

. Via the INFP utility, which initializes the FP file. This allows you 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 FP file then contains the selected entities, on which the analysis will be performed. (For information on the INFP procedure, see Subchapter 'INFP: FP File Initialization (Impact Analysis)'.)

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 CONDITION

None.

ABNORMAL EXECUTIONS

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 sought string. (Must be ended by the separator if a separator is specified)
35	1		Where the string is to be sought
		'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 sought value Required, ending with the separator (either ' or ")
35	1		Where the constant is to be sought
		'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  :    EFN : $NMTU.$ROOT$ROOTAE
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Entities in production
  PAC7FP  :    EFN : $NMBU.$ROOT$FILEFP

.Transaction file:
 -User input
  PAC7MB  :    TMBISOS

.Output file(s):
 -Selected entry points
  PAC7FH  :    TPAC7FH
 -Impact analysis results
  PAC7MF  :    TPAC7FO

.Output report(s):
  PAC7IE
```

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):
 SWK
```

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):
 SWK
```

7.3.4. ISOS: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM',
     SIZEWK=2,
     PAC7FH=$NMBU.$ROOT$FILEFH,
     PAC7FO=$NMBU.$ROOT$FILEFO,
     PAC7FQ=$NMBU.$ROOT$FILEFQ,
     PAC7FR=$NMBU.$ROOT$FILEFR;
CR   IF=*ISOS,
     OF=(TMBISOS,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PAN212 ***';
STEP PAN212,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAC7FP,$NMBU.$ROOT$FILEFP,&RFBU,
         SHARE=MONITOR;
     DEF PAC7FP,READLOCK=STAT;
     ASG PAC7MB,TMBISOS,TEMPRY,&RFTM;
     ASG PAC7FH,TPAC7FH,TEMPRY,&RFTM,END=PASS;
     ALC PAC7FH,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
     ASG PAC7MF,TPAC7FO,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MF,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PAN215 ***';
STEP PAN215,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 110;
     ASG PAC7FH,TPAC7FH,TEMPRY,&RFTM;
     ASG PAC7HF,&PAC7FH/G+1;
     ASG PAC7FR,&PAC7FR/G+1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PAN260 ***';
STEP PAN260,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 110;
     ASG PAC7MF,TPAC7FO,TEMPRY,&RFTM;
     ASG PAC7OF,&PAC7FO;
     ASG PAC7FO,&PAC7FO/G+1;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** SHIFT FILES ***';
SHIFT &PAC7FH;
SHIFT &PAC7FO;
SHIFT &PAC7FR;
JUMP END;
ERR:
SEND ' PBEXISOS - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

7.4. IMFH :MERGE FH FILES

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   :   TPAC7FH = $NMBU.$ROOT$FILEFH
                       + other to be merged

.Permanent output files:
 -Sorted selected entry points
 PAC7HF   :   $NMBU.$ROOT$FILEFH/G+1
 -Reduced entry points to be purged
 PAC7FR   :   $NMBU.$ROOT$FILEFR/G+1

.Sort file(s):
SWK

.Return codes:
```

7.4.3. IMFH: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM',
     SIZEWK=2,
     PAC7FH=$NMBU.$ROOT$FILEFH,
     PAC7FR=$NMBU.$ROOT$FILEFR;
CR   INFILES=($NMBU.$ROOT$FILEFH/G-1,$NMBU.$ROOT$FILEFH)
     OF=(TPAC7FH,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=4096,RECSZ=160);
COMM '*** PAN215 ***';
STEP PAN215,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 110;
     ASG PAC7FH,TPAC7FH,TEMPRY,&RFTM;
     ASG PAC7HF,&PAC7FH/G+1;
     ASG PAC7FR,&PAC7FR/G+1;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** SHIFT FILES ***';
SHIFT &PAC7FH;
SHIFT &PAC7FR;
```

7.5. IANA: IMPACT SEARCH CRITERIA

7.5.1. IANA: INTRODUCTION

IANA: INTRODUCTION

The IANA procedure searches for 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 IANA execution.

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 FO file contains the impact analysis cumulative results over the subsequent executions of the IANA procedure.

You may choose to reinitialize it before the first IANA execution following a new ISEP execution. If you do not reinitialize it, cumulated results will be obtained in relation to different contexts and/or entry points. Such results can then be processed globally.

The FP file, used as input for the analysis procedures, contains the specification of entities or entity types to be analyzed. If nothing is specified in this file, all analyzable entities will be analyzed.

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

For information on how to initialize the FP file, refer to the subchapter describing the INFP procedure thereafter.

The FQ file contains the already impacted criteria. It ensures that these are not impacted again by the next IANA execution. This is why this file is not automatically reinitialized.

However, it is recommended to reinitialize it before the first IANA execution following a new ISEP execution.
To reinitialize the FQ file, run the INFQ procedure (documented in Subchapter 'INFQ: FQ File Reinitialization (Impact Analysis)').

EXECUTION CONDITION

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

ABNORMAL EXECUTIONS

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, 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  :    EFN : $NMBU.$ROOT$FILEFH
 -Criteria after purge (reduced file)
  PAC7FR  :    EFN : $NMBU.$ROOT$FILEFR

.Output file:
 -Search criteria
  PAC7HF  :    EFN : TPAC7HF  Longueur=160
```

PRINTING OF ENTRY POINTS: PAN220

```
.Permanent input files:
 -Error messages
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Sorted criteria
  PAC7HF  :    EFN : TPAC7HF

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

.Sort file(s):
  SWK
```

IMPACT ANALYSIS: PAN250

```
.Permanent input files:
 -Error messages
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -File of entities to be analyzed
  PAC7FP  :    EFN : $NMBU.$ROOT$FILEFP


.Transaction file:
 -Impacted criteria
  PAC7FH  :    EFN : TPAC7HF  Length=160

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

.Output files:
 -New impacted criteria
  PAC7HF  :    EFN : TPAC7FH  Length=160
 -Impact analysis results
  PAC7MF  :    EFN : TPAC7FO  Length=260
```

UPDATE OF IMPACT ANALYSIS RESULTS: PAN260

```
.Transaction file:
 -Impact analysis results (level)
  PAC7MF  :    EFN : TPAC7FO

.Permanent input file:
 -Results of previous analysis
  PAC7OF  :    EFN : $NMBU.$ROOT$FILEFO

.Permanent output file:
 -Sorted results of impact analysis
  PAC7FO  :    EFN : $NMBU.$ROOT$FILEFO/G+1

.Sort file(s):
  SWK
```

REMOVAL OF DUPLICATE ENTRY POINTS: PAN215

```
.Transaction file:
 -Selected entry points
  PAC7FH  :    EFN : TPAC7FH

.Permanent output file:
 -Sorted selected entry points
  PAC7HF  :    EFN : $NMBU.$ROOT$FILEFH/G+1
 -Reduced entry points to be purged
  PAC7FR  :    EFN : $NMBU.$ROOT$FILEFR/G+1

.Sort file(s):
  SWK
```

7.5.3. IANA: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM',
     SIZEWK=2,
     PAC7FH=$NMBU.$ROOT$FILEFH,
     PAC7FO=$NMBU.$ROOT$FILEFO,
     PAC7FQ=$NMBU.$ROOT$FILEFQ,
     PAC7FR=$NMBU.$ROOT$FILEFR;
COMM '*** PAN230 ***';
STEP PAN230,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7FH,&PAC7FH;
     ASG PAC7FR,&PAC7FR;
     ASG PAC7HF,TPAC7HF,TEMPRY,&RFTM,END=PASS;
     ALC PAC7HF,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PAN220 ***';
STEP PAN220,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7HF,&PAC7FH;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IL,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PAN250 ***';
STEP PAN250,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAC7FP,$NMBU.$ROOT$FILEFP,&RFBU,
         SHARE=MONITOR;
     DEF PAC7FP,READLOCK=STAT;
     ASG PAC7FH,TPAC7HF,TEMPRY,&RFTM;
     ASG PAC7FQ,&PAC7FQ;
     ASG PAC7HF,TPAC7FH,TEMPRY,&RFTM,END=PASS;
     ALC PAC7HF,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
     ASG PAC7MF,TPAC7FO,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MF,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PAN260 ***';
STEP PAN260,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 110;
     ASG PAC7MF,TPAC7FO,TEMPRY,&RFTM;
     ASG PAC7OF,&PAC7FO;
```

```
     ASG PAC7FO,&PAC7FO/G+1;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PAN215 ***';
STEP PAN215,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 110;
     ASG PAC7FH,TPAC7FH,TEMPRY,&RFTM;
     ASG PAC7HF,&PAC7FH/G+1;
     ASG PAC7FR,&PAC7FR/G+1;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** SHIFT FILES ***';
SHIFT &PAC7FH;
SHIFT &PAC7FO;
SHIFT &PAC7FR;
```

7.6. IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS

7.6.1. IPIA: INTRODUCTION

IPIA: INTRODUCTION

The function of the IPIA procedure is to print reports and to format the analysis results into 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 as 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 criteria:

For each impacted field, list of entry point(s) 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 sought 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 sought 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 merged together.

>>> 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 sought by the ISOS procedure.

>>> Report requested by value '1' in Position 23 of the P-type user input line.

EXECUTION CONDITION

None, but the FO file must exist and must not be empty.

ABNORMAL EXECUTIONS

Whichever the cause of the abend is, 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		Not used
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

7.6.3. IPIA: DESCRIPTION OF STEPS

IPIA: DESCRIPTION OF STEPS

PRINTING OF IMPACT RESULTS: PAN270

```
.Permanent input files:
 -Error messages
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Impact results
  PAC7FO  :    EFN : $NMBU.$ROOT$FILEFO

.Transaction file:
 -User input
  PAC7MB  :    EFN : TMBIPIA

.Output file:
 -Generated batch transactions
  PAC7MV  :    EFN : TPAC7MV  Longueur=80

.Output report:
 -Analysis results
  PAC7IF

.Sort file(s):
  SWK
```

PRINTING OF GENERATED TRANSACTIONS: PAN280

```
.Permanent input files:
 -Error messages
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE

.Transaction file:
 -User input
  PAC7MB  :    EFN : TMBIPIA
 -Generated batch transactions
  PAC7MV  :    EFN : TPAC7MV

.Output files:
 -Selected batch transactions
  PAC7VM  :    EFN : TPAC7VM  Longueur=80

.Output report
 -List of transactions by library
  PAC7IT
```

7.6.4. IPIA: EXECUTION JCL

```
MVL  USER=$USER,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM',
     SIZEWK=2,
     PAC7FO=$NMBU.$ROOT$FILEFO;
CR   IF=*IPIA,
     OF=(TMBIPIA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PAN270 ***';
STEP PAN270,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7FO,&PAC7FO;
     ASG PAC7MB,TMBIPIA,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
     ASG PAC7IF,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PAN280 ***';
STEP PAN280,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7MB,TMBIPIA,TEMPRY,&RFTM;
     ASG PAC7MV,TPAC7MV,TEMPRY,&RFTM,END=PASS;
     ASG PAC7VM,TPAC7VM,TEMPRY,&RFTM,END=PASS;
     ASG PAC7IT,SYS.OUT;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** LIBMAINT ***';
LMN  SL INFILE=(TPAC7MV,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSU,&RFLI),
     COM='MV INFILE:MBUPDT_IPIA'&USER',INFORM=SARF,
          TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
COMM '*** LIBMAINT ***';
LMN  SL INFILE=(TPAC7VM,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSU,&RFLI),
     COM='MV INFILE:MBUPDT_IPIA2'&USER',INFORM=SARF,
          TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
```

7.7. IGRA: BREAKING DOWN OF GROUP FIELDS

7.7.1. IGRA: INTRODUCTION

IGRA - OVERVIEW

The IGRA procedure breaks down 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 impact search criteria.

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 (in a text 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 criteria' and 'eliminated 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 in input by this procedure. It contains a list of Entities or Entity Types which should be anlyzed. If no entry 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. (All-purpose coding with ****** is also possible.)

EXECUTION CONDITION

None, except that the FH file (entry points or impact search criteria) must exist and must not be empty.

ABNORMAL EXECUTIONS

Whatever the reason for the abnormal ending, the procedure may be resumed as it is after correcting the problem. However, you should check the status of generation files (FH, FR, and FO).

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   :   $NMBU.$ROOT$FILEFH
 -Reduced file of purged criteria
 PAC7FR   :   $NMBU.$ROOT$FILEFR

.Output file:
 -Search criteria file
```

PRINTING ENTRY POINTS: PAN220

```
.Permanent input files:
 -Error messages
 PAC7AE   :   $NMTU.$ROOT$ROOTAE
 -Sorted criteria
 PAC7HF   :   TPAC7HF

.Output reports:
 -List of accepted/eliminated criteria
 PAC7IL

.Sort files:
SWK
```

GROUP FIELD BREAKING-DOWN: PAN255

```
.Permanent input files:
 -Error messages
 PAC7AE   :   $NMTU.$ROOT$ROOTAE
 -Data file
 PAC7AR   :   $NMTU.$ROOT$FILEAR
 -Index file
 PAC7AN   :   $NMTU.$ROOT$FILEAN
 -Entities to be analyzed
 PAC7FP   :   $NMBU.$ROOT$FILEFP

.Transaction file:
 -Impacted criteria
 PAC7FH   :   TPAC7HF    Length=160

.Output file:
 -Impact analysis results
 PAC7MF   :   TPAC7FO    Length=260

.Return codes:
```

UPDATE OF IMPACT ANALYSIS RESULTS: PAN260

```
.Transaction file:
 -Impact analysis result (by level)
 PAC7MF   :   TPAC7FO

.Permanent input file:
 -Results of previous analysis
 PAC7OF   :   $NMBU.$ROOT$FILEFO
```

```
.Permanent output file:
 -Sorted results of the impact analysis
 PAC7FO   :   $NMBU.$ROOT$FILEFO/G+1

.Sort files:
SWK

.Return codes:
```

7.7.3. IGRA: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM',
     SIZEWK=2,
     PAC7FH=$NMBU.$ROOT$FILEFH,
     PAC7FO=$NMBU.$ROOT$FILEFO,
     PAC7FR=$NMBU.$ROOT$FILEFR;
COMM '*** PAN230 ***';
STEP PAN230,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7FH,&PAC7FH;
     ASG PAC7FR,&PAC7FR;
     ASG PAC7HF,TPAC7HF,TEMPRY,&RFTM,END=PASS;
     ALC PAC7HF,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
COMM '*** PAN220 ***';
STEP PAN220,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7HF,TPAC7HF,TEMPRY,&RFTM,END=PASS;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IL,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
ESTP;
COMM '*** PAN255 ***';
STEP PAN255,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAC7FP,$NMBU.$ROOT$FILEFP,&RFBU,
         SHARE=MONITOR;
     DEF PAC7FP,READLOCK=STAT;
     ASG PAC7FH,TPAC7HF,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MF,TPAC7FO,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MF,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
ESTP;
COMM '*** PAN260 ***';
STEP PAN260,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ 110;
     ASG PAC7MF,TPAC7FO,TEMPRY,&RFTM;
     ASG PAC7OF,&PAC7FO;
     ASG PAC7FO,&PAC7FO/G+1;
ESTP;
COMM '*** SHIFT FILES ***';
SHIFT &PAC7FH;
SHIFT &PAC7FO;
SHIFT &PAC7FR;
```

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 (or not used) 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 searche 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-Elements defined in the Dictionary
. Data-Elements defined in Segment Descriptions
. Data-Elements defined in Report Structures
. Data-Elements 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 CONDITION

None, but the FQ file must exist.
Run the PBININIT procedure with 'VL=FH' as parameter in order to initialize the FH file.

ABNORMAL EXECUTIONS

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 area
		'G'	For a Group area
		' '	For an elementary area
		'*'	For all types of areas

7.8.3. IPFQ: DESCRIPTION OF STEPS

IPFQ: DESCRIPTION OF STEPS

EXTRACTION OF CRITERIA: PAN240

```
.Permanent input files:
 -Error messages
  PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Data file
  PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Index file
  PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Criteria impacted during analysis
  PAC7FQ  :    EFN : $NMBU.$ROOT$FILEFQ

.Transaction file:
 -Input
  PAC7MB  :    TMBIPFQ

.Output files:
 -Search criteria
  PAC7FH

.Output report:
 -Control report
  PAC7IX
```

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):
 SWK
```

7.8.4. IPFQ: EXECUTION JCL

```
MVL  CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM',
     SIZEWK=2,
     PAC7FQ=$NMBU.$ROOT$FILEFQ;
CR   IF=&PAC7FQ,
     OF=(TPAC7FQ,&RFTM,TEMPRY,END=PASS),
     OUTDEF=(CISZ=4096,RECSZ=100,RECFORM=FB);
COMM '*** PAN240 ***';
STEP PAN240,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$FILEAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AN,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PAC7FQ,TPAC7FQ,TEMPRY,&RFTM,END=PASS;
     ASG PAC7FH,TPAC7FH,TEMPRY,&RFTM,END=PASS;
     ALC PAC7FH,SZ=&SIZEWK,UNIT=CYL,INCRSZ=2;
     ASG PAC7EI,SYS.OUT;
     ASG PAC7IX,SYS.OUT;
     ASG PAC7MB,TMBIPFQ,TEMPRY,&RFTM,END=PASS;
ESTP;
JUMP ERR,SW20,EQ,1;
COMM '*** PAN220 ***';
STEP PAN220,FILE=($NMLI.$LIBLM,&RFLI),REPEAT,DUMP=DATA;
     SZ  160;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7HF,TPAC7FH,TEMPRY,&RFTM,END=PASS;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IL,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' PBEXIPFQ - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

7.9. INIT: USER-FILE REINITIALIZATION (IMPACT ANAL.)

7.9.1. INIT: INTRODUCTION

INITIALIZATION OF USER FILES

Execute the PBININIT procedure:

```
S: EJ PBININIT,,$NMLI.$LIBJCL VL=$1

   where $NMLI.$LIBJCL is the name of the VA Pac JCL
library

   If no file name is indicated in the value, FH, FO, FP,
   FP and FR files are set by the procedure.

   To have only one file set, specify &1=FH or &1=FO or
   &1=FQ or &1=FR as parameter.
```

The PBININIT procedure creates the user's work files (FH, FO, FQ, and FR), and initializes the FP file.

These files are catalogued.

NOTE: If you wish to limit the impact analysis to some entities, insert input entities between '$IN INIT;' and '$EIN INIT' lines of the JCL (refer to Pac/impact INFP Procedure - Reference Manual).

7.9.2. INIT: EXECUTION JCL

```
COMM 'VISUALAGE PACBASE 2.5';
MVL  ALL,INIT=&1,LABALL=ALL,JLAB='&LAB'&INIT,
     PAC7FH='$NMBU.$ROOT$FILEFH',SIZEFH=3,
     PAC7FO='$NMBU.$ROOT$FILEFO',SIZEFO=3,
     PAC7FR='$NMBU.$ROOT$FILEFR',SIZEFR=3,
     PAC7FQ='$NMBU.$ROOT$FILEFQ',SIZEFQ=3,
     PAC7FP='$NMBU.$ROOT$FILEFP',SIZEFP=3,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     DVD='DVC=$DVBU',MDD='MD=$MDBU';
JUMP &INIT;
ALL:
FH:
COMM 'ALLOCATION : FH';
DALC &PAC7FH;
JUMP CONTINUE;
DALC &PAC7FH!!/G-1;
JUMP CONTINUE;
UNCAT &PAC7FH,TYPE=FILE;
JUMP CONTINUE;
CAT  &PAC7FH,TYPE=FILE,NBGEN=2,SHARE=UNSPEC;
PALC &PAC7FH/G+1,
     UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFH),INCRSZ=1,
     UFAS=(SEQ=(CISZ=4096,RECSZ=160)),
     FILESTAT=CAT;
PALC &PAC7FH/G+1,
     UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFH),INCRSZ=1,
     UFAS=(SEQ=(CISZ=4096,RECSZ=160)),
     FILESTAT=CAT;
JUMP EFH&&JLAB;
EFH: JUMP END;
EFHALL:
FO:
COMM 'ALLOCATION : FO';
DALC &PAC7FO;
JUMP CONTINUE;
DALC &PAC7FO!!/G-1;
JUMP CONTINUE;
UNCAT &PAC7FO,TYPE=FILE;
JUMP CONTINUE;
CAT  &PAC7FO,TYPE=FILE,NBGEN=2,SHARE=UNSPEC;
PALC &PAC7FO/G+1,
     UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFO),INCRSZ=1,
     UFAS=(SEQ=(CISZ=4096,RECSZ=260)),
     FILESTAT=CAT;
PALC &PAC7FO/G+1,
     UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFO),INCRSZ=1,
     UFAS=(SEQ=(CISZ=4096,RECSZ=260)),
     FILESTAT=CAT;
JUMP EFO&&JLAB;
EFO: JUMP END;
EFOALL:
FR:
COMM 'ALLOCATION : FR';
DALC &PAC7FR;
JUMP CONTINUE;
DALC &PAC7FR!!/G-1;
JUMP CONTINUE;
UNCAT &PAC7FR,TYPE=FILE;
JUMP CONTINUE;
CAT  &PAC7FR,TYPE=FILE,NBGEN=2,SHARE=UNSPEC;
PALC &PAC7FR/G+1,
     UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFR),INCRSZ=1,
     UFAS=(SEQ=(CISZ=4096,RECSZ=72)),
     FILESTAT=CAT;
PALC &PAC7FR/G+1,
     UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFR),INCRSZ=1,
```

```
         UFAS=(SEQ=(CISZ=4096,RECSZ=72)),
         FILESTAT=CAT;
    JUMP EFR&&JLAB;
    EFR: JUMP END;
    EFRALL:
    FQ:
    COMM 'ALLOCATION : FQ';
    DALC &PAC7FQ;
    JUMP CONTINUE;
    UNCAT &PAC7FQ,TYPE=FILE;
    JUMP CONTINUE;
    CAT  &PAC7FQ,TYPE=FILE,SHARE=UNSPEC;
    PALC &PAC7FQ,
         UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFQ),INCRSZ=1,
         UFAS=(INDEXED=(CISZ=4096,RECSZ=100,KEYLOC=2,KEYSZ=94)),
         FILESTAT=CAT;
    COMM '*** PAN200 ***';
    STEP PAN200,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
         SZ  130;
         ASG PAC7FQ,&PAC7FQ;
    ESTP;
    JUMP EFQ&&JLAB;
    EFQ: JUMP END;
    EFQALL:
    FP:
    $IN INIT;
    $EIN INIT;
    CR   IF=*INIT,
         OF=(TMBINIT,TEMPRY,&RFTM,END=PASS),
         OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
    COMM 'ALLOCATION : FP';
    DALC &PAC7FP;
    JUMP CONTINUE;
    UNCAT &PAC7FP,TYPE=FILE;
    JUMP CONTINUE;
    CAT  &PAC7FP,TYPE=FILE,SHARE=UNSPEC;
    PALC &PAC7FP,
         UNIT=CYL,&DVD,GBL=(&MDD,SZ=&SIZEFP),INCRSZ=1,
         UFAS=(INDEXED=(CISZ=4096,RECSZ=9,KEYLOC=1,KEYSZ=9)),
         FILESTAT=CAT;
    COMM '*** PAN205 ***';
    STEP PAN205,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
         SZ  130;
         ASG PAC7MB,TMBINIT,TEMPRY,&RFTM;
         ASG PAC7FP,&PAC7FP;
         ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
             SHARE=MONITOR;
         DEF PAC7AE,READLOCK=STAT;
         ASG PAC7IP,SYS.OUT;
    ESTP;
    JUMP EFP&&JLAB;
    EFP: JUMP END;
    EFPALL:
    END:
```

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 Proxy Logical Views used in the extracted objects. It can be used to re-generate the Proxy Logical Views if needed.

EXECUTION CONDITION

None.

ABNORMAL EXECUTIONS

For details on the abnormal executions, see the Manual: 'Batch procedures: Administrator's Guide', Chapter 'OVERVIEW', Subchapter 'Abnormal Endings'.

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	SPACE	Session number and status Current session	O

(*) R = Required, O = Optional

2. Extraction command line (one line per object)

Pos.	Len.	Value	Meaning	(*)
2	2	'Y3'	Line code	R
4	2	'77'	Object's class VisualAge Smltlk. application	R
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  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Error messages
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -User input
   PAC7MB  :    EFN : TMBVDWN

.Output reports and files:
 -Check report
   PAC7ET
 -'*'-line check report
   PAC7DD
 -Proxy-generation requests (GPRT)
   (length: 80)

This file will 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 for the GPRT procedure.   PAC7ME  :
EFN=TMBGPRT (parameter of the procedure)
 -Elementary-extraction requests
   PAC7MV  :    EFN : TMVVDWN
```

EXTRACTION: PVA110

```
.Input files:
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Error messages
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Elementary extraction requests
   PAC7MV  :    EFN : TMVVDWN

.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 VA Smalltalk environment.   PAC7MX  :
EFN=$NMBU.VDWN.&USER
```

8.1.4. VDWN: EXECUTION JCL

```
MVL  BVVISU='$NMBU.VDWN_$USER',USER='$USER',SIZEVI=1,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*VDWN,
     OF=(TMBVDWN,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
COMM '*** PVA100 ***';
STEP PVA100,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7MB,TMBVDWN,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7MV,TMVVDWN,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7ME,TMVGPRT,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ME,NBBUF=1;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
LMN  SL INFILE=(TMVGPRT,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSU,&RFLI),
     COM='MV INFILE:MVGPRT_VDWN'&USER',INFORM=SARF,
          TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
COMM '*** ALLOCATION FICHIER VS ***';
IV PBINALVI ($NMLI.$LIBJCL,&RFLI) VL=(&SIZEVI,&USER);
COMM '*** PVA110 ***';
STEP PVA110,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7EI,SYS.OUT;
     ASG PAC7MV,TMVVDWN,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MX,&BVVISU,&RFBU;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
```

8.2. VUP1: BACKUP - CODE CALCULATION

8.2.1. VUP1: INTRODUCTION

VUP1: INTRODUCTION

This procedure creates the elements which will be used as input by the VUP2 procedure to generate the backup transactions in VisualAge Pacbase. These transactions will be used as input by the UPDT procedure.

The VUP1 procedure creates three files:

1. 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 entities created in VisualAge Smalltalk/Java. 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.

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTIONS

For details on the abnormal executions, see the 'Batch Procedures: Administrator's Guide', Chapter OVERVIEW, Subchapter 'Abnormal Endings'.

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 VisualAge Smalltalk/ Java environment. 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 '*' 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 changed.

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

The administrator has to full the BVTBCHAR table in the $LIBSRT library.

8.2.3. VUP1: DESCRIPTION OF STEPS

VUP1: DESCRIPTION OF STEPS

EXTRACTION OF VISUALAGE SMALLTALK/JAVA ENTITY CODES FROM VISUALAGE PACBASE: PVA300

```
.Input files:
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Error messages
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -VisualAge Smalltalk/Java file produced by workstation
   PAC7VA       &BVVISU

.Output reports and files:
 -Check report
   PAC7ET
 -'*'-line check report
   PAC7DD
 -Extracted codes
   PAC7VC  :    TPAC7VC

.Sort file(s):
   SWK
```

COMPARISON OF ENTITIES EXTRACTED FROM VA PAC AND NEW ENTITIES TO BE CREATED IN VA PAC: PVA305

```
.Input files:
 -Index file
   PAC7AN  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Error message file
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -VisualAge Smalltalk/Java file produced by the workstation
   PAC7VA  :    &BVVISU
 -VisualAge Pacbase codes of VisualAge Smalltalk/Java
  entities already saved
   PAC7VC  :    TPAC7VC


  PAC7CA   EFN  :   TPAC7CA


.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  :        &BVPBCO
 -Useful VisualAge Smalltalk/Java transactions
   PAC7VG  :        &BVVUTI

.Sort file(s):
   SWK
```

CALCULATION OF VA PAC CODES FOR NEW VA SMALLTALK/JAVA
ENTITIES: PVA310

.Input files:
-Index file
PAC7AN : EFN : $NMTU.$ROOT$FILEAN
-Data file
PAC7AR : EFN : $NMTU.$ROOT$FILEAR
-Error message file
PAC7AE : EFN : $NMTU.$ROOT$ROOTAE
-VisualAge Smalltalk/Java file produced by the workstation
PAC7VA : &BVVISU
-VisualAge Pacbase codes of VisualAge Smalltalk/Java
entities already saved
PAC7VC : TPAC7VC

-Character-correspondence table
for substitution in the code calculation
PAC7CA : file to "override"

.Output reports and file:
-List of new codes created
PAC7ET
-'*'-line check report
PAC7DD
-List of codes assigned to new VisualAge Smalltalk/Java
entities
PAC7VN : &BVPBCO
PAC7VG : &BVVUTI (paramètre de la procédure)
-List of VisualAge Pacbase codes of VisualAge Smalltalk/
Java entities already saved
PAC7VC : TPAC7VC
-List of codes assigned to the new VisualAge Smalltalk/Java
entities
PAC7VV : &PBCOD (paramètre de la procédure)
-File of codes assigned to entities already stored in
VisualAge Pacbase
PAC7VP : EFN : $NMTU.$ROOT$FILEVP

.Sort files:
SWK

8.2.4. VUP1: EXECUTION JCL

```
COMM '*******************************************************';
COMM '*         - VISUALAGE SMALLTALK BRIDGE -              *';
COMM '*           UPLOAD:  NEW PACBASE CODES                *';
COMM '*******************************************************';
COMM '* BVVISU          FILE FROM VA SMALLTALK               *';
COMM '* BVVUTI,         FILE FROM VA SMALLTALK WITH ONLY     *';
COMM '*                            USEFUL TRANSACTIONS      *';
COMM '* BVPBCO          FILE OF NEW VA PAC CODES TO MODIFY   *';
COMM '*******************************************************';
MVL  BVVISU='$NMBU.VISU_$USER',SIZEVC=1,
     BVPBCO='$NMBU.PBCO_$USER',USER='$USER',SIZECO=1,
     BVVUTI='$NMBU.VUTI_$USER',SIZEVU=1,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
COMM '*** ALLOCATION FICHIER VP DES CODES DEJA STOCKES ***';
IV PBINALVP ($NMLI.$LIBJCL,&RFLI);
COMM '*** ALLOCATION FICHIER VP DES NOUVEAUX CODES     ***';
COMM '*** PVA300 ***';
STEP PVA300,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7VA,&BVVISU,&RFBU;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7VC,TPAC7VC,TEMPRY,&RFTM,END=PASS;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
CR:
CR   IF=($NMLI.$LIBSRT,&RFLI,SUBFILE=BVTBCHAR),
     OF=(TPAC7CA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB),
     COMFILE=($NMLI.$LIBJCL,&RFLI,SUBFILE=PBEXPDSL),START=2;
CREND:
COMM '*** ALLOCATION FICHIER CO ***';
IV PBINALCO ($NMLI.$LIBJCL,&RFLI) VL=(&SIZECO,&USER);
COMM '*** ALLOCATION FICHIER VU ***';
IV PBINALVU ($NMLI.$LIBJCL,&RFLI) VL=(&SIZEVU,&USER);
COMM '*** PVA305 ***';
STEP PVA305,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
```

```
        ASG PAC7VA,&BVVISU,&RFBU;
        ASG PAC7VC,TPAC7VC,TEMPRY,&RFTM;
        ASG PAC7ET,SYS.OUT;
        ASG PAC7DD,SYS.OUT;
        ASG PAC7ED,SYS.OUT;
        ASG PAC7VN,&BVPBCO,&RFBU;
        ASG PAC7VG,&BVVUTI,&RFBU;
   ESTP;
   JUMP ERR,SW20,EQ,1;
   JUMP END,SW30,EQ,1;
   COMM '*** PVA310 ***';
   STEP PVA310,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
        SZ  120;
        ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AE,READLOCK=STAT;
        ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AN,NBBUF=1,READLOCK=STAT;
        ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
            ACC=READ,SHARE=MONITOR;
        DEF PAC7AR,NBBUF=1,READLOCK=STAT;
        ASG PAC7CA,TPAC7CA,TEMPRY,&RFTM;
        ASG PAC7VA,&BVVISU,&RFBU;
        ASG PAC7VC,TPAC7VC,TEMPRY,&RFTM;
        ASG PAC7ET,SYS.OUT;
        ASG PAC7DD,SYS.OUT;
        ASG PAC7ED,SYS.OUT;
        ASG PAC7VN,&BVPBCO,&RFBU;
        ASG PAC7VV,&BVVUTI,&RFBU;
        ASG PAC7VP,$NMBU.$ROOT$FILEVP,&RFBU;
   ESTP;
   JUMP ERR,SW20,EQ,1;
   JUMP END,SW30,EQ,1;
```

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 EXECUTIONS

For details on the abnormal executions, see the Administrator's Guide, Chapter 'OVERVIEW', Subchapter 'Abnormal Endings'.

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 to generate the update transactions of the VisualAge Pacbase database.

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
11	8		Password	R
58	9		Product + Change number if database under DSMS control	O

(*) R = Required, O = Optional

2. The file of MODIFIED VA Pac 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 modifications.

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  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR
 -Error messages
   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -Useful transactions produced by VisualAge Smalltalk
  (from the workstation)
   PAC7VA  :    EFN : $BVVUTI
  (&VISUTIL file produced by VUP1)
 -Codes of new VisualAge Smalltalk/Java entities taken into
  account
   PAC7VN  :    EFN : $BVPBCO
  (&PBCOD file produced by VUP1)
 -Codes of VisualAge Smalltalk/Java entities already saved
  in VisualAge Pacbase
   PAC7VP  :    EFN : $NMBU.$ROOT$FILEVP

.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:
  PAC7VC
 -Transactions for UPDT that include only definitions
   PAC7MY  :   TPAC7MY
 -Transactions for UPDT other than definitions
   PAC7MX  :   TPAC7MX

.Sort file(s):
   SWK
```

8.3.4. VUP2: EXECUTION JCL

```
MVL  BVPBCO='$NMBU.PBCO_$USER',USER='$USER',SIZECO=1,
     BVVUTI='$NMBU.VUTI_$USER',USER='$USER',SIZEVU=1,
     CTTUN='FILESTAT=UNCAT,DVC=$DVTU,MD=$MDTU',
     RFTU=&CTTU$CTTU,
     CTBSN='FILESTAT=UNCAT,DVC=$DVBS,MD=$MDBS',
     RFBS=&CTBS$CTBS,
     CTLIN='FILESTAT=UNCAT,DVC=$DVLI,MD=$MDLI',
     RFLI=&CTLI$CTLI,
     CTBUN='FILESTAT=UNCAT,DVC=$DVBU,MD=$MDBU',
     RFBU=&CTBU$CTBU,
     CTAJN='FILESTAT=UNCAT,DVC=$DVAJ,MD=$MDAJ',
     RFAJ=&CTAJ$CTAJ,
     RFTM='DVC=$DVTM,MD=$MDTM';
COMM '*** PVA320 ***';
STEP PVA320,FILE=($NMLI.$LIBLM,&RFLI),DUMP=DATA;
     SZ  120;
     ASG PAC7AE,$NMTU.$ROOT$ROOTAE,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$NMTU.$ROOT$FILEAN,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=1,READLOCK=STAT;
     ASG PAC7AR,$NMTU.$ROOT$FILEAR,&RFTU,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=1,READLOCK=STAT;
     ASG PAC7VA,&BVVUTI,&RFBU;
     ASG PAC7VP,$NMBU.$ROOT$FILEVP,&RFBU;
     ASG PAC7VN,&BVPBCO,&RFBU;
     ASG PAC7ET,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7EM,SYS.OUT;
     ASG PAC7ER,SYS.OUT;
     ASG PAC7MX,TPAC7MX,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MY,TPAC7MY,TEMPRY,&RFTM,END=PASS;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
LMN  SL INFILE=(TPAC7MX,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSU,&RFLI),
     COM='MV INFILE:MBUPDT_VUP2'&USER',INFORM=SARF,
          TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
LMN  SL INFILE=(TPAC7MY,TEMPRY,&RFTM),
     LIB=($NMLI.$LIBSU,&RFLI),
     COM='MV INFILE:MBUPDT_VUP2B'&USER',INFORM=SARF,
          TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
```

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 CONDITION

None.

ABNORMAL EXECUTION

For details on the abnormal executions, see the Manual 'Batch Procedures : Administrator's Guide', Chapter 'OVERVIEW', Subchapter 'Abnormal Endings'.

8.4.2. VPUR: USER INPUT

VPUR: USER INPUT

1. VisualAge Pacbase Manager definition line:

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 will be 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 typeis entered, all sessions will be 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  :    EFN : $NMTU.$ROOT$FILEAN
 -Data file
   PAC7AR  :    EFN : $NMTU.$ROOT$FILEAR


   PAC7AE  :    EFN : $NMTU.$ROOT$ROOTAE
 -User input
   PAC7MB  :    TMBVPUR

.Output reports and file:
 -List of user input
   PAC7ET
 -'*'-line check report
   PAC7DD
 -Generated purge-transactions
   PAC7MX  :   MVVPUR_VPUR'&USER'

.Sort file(s):
   SWK
```