



VisualAge Pacbase 2.5

**VA PAC 2.5 - BULL GCOS8
OPERATIONS MANUAL VOLUME III : USER'S GUIDE**

DEPD8003251A

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.

© Copyright International Business Machines Corporation 1983, 1999. All rights reserved.

Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

NOTICES

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

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

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

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

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

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

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

TRADEMARKS

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

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

TABLE OF CONTENTS

1. GENERAL INTRODUCTION TO THE BATCH PROCEDURES	9
1.1. PRESENTATION OF THE BATCH PROCEDURES USER'S GUIDE	10
1.2. OVERVIEW OF THE PROCEDURES	11
1.3. USER IDENTIFICATION (*).....	12
2. STANDARD PROCEDURES.....	14
2.1. UPDT: DATABASE UPDATE.....	15
2.1.1. UPDT: INTRODUCTION.....	15
2.1.2. UPDT: UPDATE RULES - RESULTS.....	17
2.1.3. UPDT: DESCRIPTION OF STEPS.....	21
2.1.4. UPDT: EXECUTION JCL.....	23
2.2. UPDP: DATABASE UPDATE FROM PAF TABLES.....	25
2.2.1. UPDP: INTRODUCTION	25
2.2.2. UPDP: INPUT - PROCESSING - RESULTS	26
2.2.3. UPDP: DESCRIPTION OF STEPS.....	28
2.2.4. UPDP: EXECUTION JCL.....	30
2.3. GPRT: GENERATION AND PRINTING	32
2.3.1. GPRT: INTRODUCTION.....	32
2.3.2. GPRT: STRUCTURE OF REQUESTS	36
2.3.3. GPRT: GENERATION/PRINTING COMMANDS.....	39
2.3.4. GPRT: USER INPUT AND RESULTS.....	61
2.3.5. GPRT: DESCRIPTION OF STEPS	64
2.3.6. GPRT: EXECUTION JCL	79
2.3.7. INTERFACE WITH GDT-PC.....	109
2.3.8. EMLD: LOADING OF USER-DEFINED ERROR MESSAGES	110
2.3.8.1. EMLD: INTRODUCTION.....	110
2.3.9. EMLD: DESCRIPTION OF STEPS	111
2.3.10. EMLD: EXECUTION JCL.....	112
2.3.11. EMUP: UPDATE OF USER-DEFINED ERROR MESSAGES	113
2.3.11.1. EMUP: INTRODUCTION.....	113
2.3.12. EMUP: USER INPUT.....	114
2.3.13. EMUP: DESCRIPTION OF STEPS	115
2.3.14. EMUP: EXECUTION JCL	116
2.3.15. PPAF: PAF PRE-PROCESSOR.....	117
2.3.15.1. PPAF: INTRODUCTION	117
2.3.16. PPAF: USER INPUT.....	118
2.3.17. PPAF: DESCRIPTION OF STEPS.....	119
2.3.18. PPAF: EXECUTION JCL.....	121
2.4. PACX: EXTRACTION FROM THE VA PAC DATABASE	122
2.4.1. PACX: INTRODUCTION.....	122
2.4.2. PACX: USER INPUT COMMON TO ALL EXTRACTORS	123
2.4.3. EXLI: LIBRARY EXTRACTION	125
2.4.3.1. EXLI: INTRODUCTION.....	125
2.4.4. EXLI: USER INPUT.....	126
2.4.5. EXTR: ENTITY EXTRACTION.....	127
2.4.5.1. EXTR: INTRODUCTION	127
2.4.6. EXTR: USER INPUT.....	128
2.4.7. EXPJ: TRANSACTION EXTRACTION FROM THE JOURNAL.....	130
2.4.7.1. EXPJ: INTRODUCTION.....	130
2.4.8. EXPJ: USER INPUT.....	131
2.4.9. EXPU: EXTRACTION OF UNUSED ENTITIES FOR PURGE.....	132
2.4.9.1. EXPU: INTRODUCTION	132
2.4.10. EXPU: USER INPUT	134
2.4.11. EXUE: EXTRACTION OF USER ENTITIES.....	137
2.4.11.1. EXUE: INTRODUCTION	137
2.4.12. EXUE: USER INPUT	138
2.4.13. RMEN: RENAME/MOVE OF ENTITIES	139
2.4.13.1. RMEN: INTRODUCTION	139

2.4.14. RMEN: USER INPUT.....	140
2.4.15. RMEN: RECOMMENDATIONS AND RESTRICTIONS	144
2.4.16. PACX: DESCRIPTION OF STEPS	148
2.4.17. PACX: EXECUTION JCL	150
3. PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION.....	152
3.1. XPAF: EXTRACTION MASTER PATH.....	153
3.1.1. XPAF: INTRODUCTION	153
3.1.2. XPAF: USER INPUT.....	155
3.1.3. XPAF: DESCRIPTION OF STEPS.....	156
3.1.4. XPAF: EXECUTION JCL.....	158
3.1.5. XPAF: CREATION OF A MACRO-COMMAND	160
3.1.6. XPAF: CREATION OF A GENERAL-PURPOSE EXTRACTOR	164
3.2. XPDM: MASTER OUTLINE	169
3.2.1. XPDM: INTRODUCTION.....	169
3.2.2. XPDM: USER INPUT.....	170
3.2.3. XPDM: DESCRIPTION OF STEPS	171
3.2.4. XPDM: EXECUTION JCL	172
3.3. PAFX: GENERAL-PURPOSE EXTRACTOR.....	174
3.3.1. PAFX: INTRODUCTION	174
3.3.2. PAFX: USER INPUT.....	175
3.3.3. PAFX: DESCRIPTION OF STEPS, JCL.....	176
3.4. PRGS: PRINTING OF MASTER PATH / OUTLINE FILE	182
3.4.1. PRGS: INTRODUCTION	182
3.4.2. PRGS: USER INPUT.....	183
3.4.3. PRGS: DESCRIPTION OF STEPS.....	184
3.4.4. PRGS: EXECUTION JCL.....	185
4. QUALITY ANALYSIS AND CONTROL.....	187
4.1. ACTI: JOURNAL STATISTICS UTILITY	188
4.1.1. ACTI: INTRODUCTION	188
4.1.2. ACTI: COMMAND LANGUAGE	189
4.1.3. ACTI: USER INPUT.....	201
4.1.4. ACTI: DESCRIPTION OF STEPS.....	202
4.1.5. ACTI: EXECUTION JCL.....	203
4.2. PQC-: PACBENCH QUALITY CONTROL	204
4.2.1. PQC: INTRODUCTION.....	204
4.2.2. PQCA: QUALITY ANALYSIS	205
4.2.2.1. PQCA: INTRODUCTION	205
4.2.3. PQCA: DESCRIPTION OF STEPS.....	207
4.2.4. PQCA: EXECUTION JCL.....	208
4.2.5. PQCE: EXTRACTION OF USER-DEFINED QUALITY RULES.....	211
4.2.5.1. PQCE: INTRODUCTION.....	211
4.2.6. PQCE: USER INPUT	212
4.2.7. PQCE: DESCRIPTION OF STEPS.....	214
4.2.8. PQCE: EXECUTION JCL.....	217
5. METHODOLOGY INTEGRITY CHECK.....	219
5.1. ADM: SSADM PACDESIGN METHODOLOGY	220
5.1.1. SADM: INTRODUCTION	220
5.1.2. SADM: USER INPUT.....	221
5.1.3. SADM: DESCRIPTION OF STEPS.....	222
5.1.4. SADM: EXECUTION JCL.....	223
5.2. YSM: WORKSTATION / YSM METHODOLOGY	224
5.2.1. YSMC: INTRODUCTION.....	224
5.2.2. YSMC: USER INPUT	225
5.2.3. YSMC: DESCRIPTION OF STEPS	227
5.2.4. YSMC: EXECUTION JCL.....	229
6. PACTABLES	231

6.1. GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR.....	232
6.1.1. GETD - GETA: INTRODUCTION	232
6.1.2. GETD - GETA: USER INPUT.....	234
6.1.4. GETD: EXECUTION JCL.....	237
6.1.5. GET2 - GET1: INTRODUCTION.....	239
6.1.6. GET2 - GET1: USER INPUT.....	241
6.1.7. GET2 - GET1: DESCRIPTION OF STEPS.....	243
6.1.8. GET2: EXECUTION JCL.....	244
6.2. GET1-GET0: INITIALIZATION OF DESCRIPTION FILE.....	245
6.2.1. GET1: INTRODUCTION.....	245
6.2.2. GET1: DESCRIPTION OF STEPS.....	246
6.2.3. GET1: EXECUTION JCL.....	247
6.2.4. GET0: INTRODUCTION.....	248
6.2.5. GET0: DESCRIPTION OF STEPS.....	249
6.2.6. GET0: EXECUTION JCL.....	250
7. PAC/IMPACT.....	251
7.1. ISEP: SELECTION OF ENTRY POINTS.....	253
7.1.1. ISEP: INTRODUCTION.....	253
7.1.2. ISEP: USER INPUT.....	255
7.1.3. ISEP: DESCRIPTION OF STEPS.....	257
7.1.4. ISEP: EXECUTION JCL.....	258
7.2. IPEP: ENTRY-POINT PRINTOUT.....	260
7.2.1. IPEP: INTRODUCTION.....	260
7.2.2. IPEP: DESCRIPTION OF STEPS.....	261
7.2.3. IPEP: EXECUTION JCL.....	262
7.3. ISOS: SELECTION OF STRINGS AND OPERATORS.....	263
7.3.1. ISOS: INTRODUCTION.....	263
7.3.2. ISOS: USER INPUT.....	265
7.3.3. ISOS: DESCRIPTION OF STEPS.....	267
7.3.4. ISOS: EXECUTION JCL.....	268
7.4. IMFH : MERGE FH FILES.....	270
7.4.1. IMFH: INTRODUCTION.....	270
7.4.2. IMFH: DESCRIPTION OF STEPS.....	271
7.4.3. IMFH: EXECUTION JCL.....	272
7.5. IANA: IMPACT SEARCH CRITERIA.....	273
7.5.1. IANA: INTRODUCTION.....	273
7.5.2. IANA: DESCRIPTION OF STEPS.....	275
7.5.3. IANA: EXECUTION JCL.....	277
7.6. IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS.....	279
7.6.1. IPIA: INTRODUCTION.....	279
7.6.2. IPIA: USER INPUT.....	282
7.6.3. IPIA: DESCRIPTION OF STEPS.....	284
7.6.4. IPIA: EXECUTION JCL.....	285
7.7. IGRA: BREAKING DOWN OF GROUP FIELDS.....	287
7.7.1. IGRA: INTRODUCTION.....	287
7.7.2. IGRA: DESCRIPTION OF STEPS.....	289
7.7.3. IGRA: EXECUTION JCL.....	291
7.8. IPFQ: FQ FILE PRINTOUT (IMPACT ANALYSIS).....	293
7.8.1. IPFQ: INTRODUCTION.....	293
7.8.2. IPFQ: USER INPUT.....	294
7.8.3. IPFQ: DESCRIPTION OF STEPS.....	295
7.8.4. IPFQ: EXECUTION JCL.....	296
7.9. INFQ: FQ FILE REINITIALIZATION (IMPACT ANALYSIS).....	297
7.9.1. INFQ: INTRODUCTION.....	297
7.9.2. INFQ: DESCRIPTION OF STEPS.....	298
7.9.3. INFQ: EXECUTION JCL.....	299
7.10. INFP: FP FILE INITIALIZATION (IMPACT ANALYSIS).....	300
7.10.1. INFP: INTRODUCTION.....	300
7.10.2. INFP: USER INPUT.....	301

7.10.3. INFP: DESCRIPTION OF STEPS	302
7.10.4. INFP: EXECUTION JCL.....	303
8. VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE.....	304
8.1. VDWN: RESTORATION.....	305
8.1.1. VDWN: INTRODUCTION.....	305
8.1.2. VDWN: USER INPUT	306
8.1.3. VDWN: DESCRIPTION OF STEPS.....	307
8.1.4. VDWN: EXECUTION JCL.....	308
8.2. VUP1: BACKUP - CODE CALCULATION.....	309
8.2.1. VUP1: INTRODUCTION	309
8.2.2. VUP1: USER INPUT.....	311
8.2.3. VUP1: DESCRIPTION OF STEPS.....	313
8.2.4. VUP1: EXECUTION JCL.....	315
8.3. VUP2: GENERATION OF UPDT TRANSACTIONS.....	317
8.3.1. VUP2: INTRODUCTION	317
8.3.2. VUP2: USER INPUT.....	318
8.3.3. VUP2: DESCRIPTION OF STEPS.....	319
8.3.4. VUP2: EXECUTION JCL.....	320
8.4. VPUR: PURGE	321
8.4.1. VPUR: INTRODUCTION.....	321
8.4.2. VPUR: USER INPUT	322
8.4.3. VPUR: DESCRIPTION OF STEPS.....	323
8.4.4. VPUR: EXECUTION JCL.....	324

VisualAge Pacbase - Operations Manual
BATCH PROCEDURES: USER'S GUIDE
GENERAL INTRODUCTION TO THE BATCH PROCEDURES

PAGE 9

1

1. GENERAL INTRODUCTION TO THE BATCH PROCEDURES

GENERAL INTRODUCTION TO THE BATCH PROCEDURES	PAGE	10
PRESENTATION OF THE BATCH PROCEDURES USER'S GUIDE		1
		1

1.1. PRESENTATION OF THE BATCH PROCEDURES USER'S GUIDE

FOREWORD

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

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

They also include the procedures dedicated to the following functionalities:

- . Personalized extraction and automated documentation
- . Quality analysis and control (PQC)
- . Integrity checks on Methodology occurrences (associated with the VA Pac WorkStation's Pacdesign module for SSADM and YSM)
- . Pactables
- . Pac/Impact
- . VisualAge Java/Smalltalk <> VisualAge Pacbase Interface

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.

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

The user, for example, must have authorization 4 to manage the Database (MLIB, REST, etc.), and authorization 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 on-
!     !    !           ! ly if you input 'C' in position 1.
-----

```

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

VisualAge Pacbase - Operations Manual
BATCH PROCEDURES: USER'S GUIDE
STANDARD PROCEDURES

PAGE 14

2

2. STANDARD PROCEDURES

STANDARD PROCEDURES	PAGE	15
UPDT: DATABASE UPDATE		2
UPDT: INTRODUCTION		1

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.

STANDARD PROCEDURES	PAGE	16
UPDT: DATABASE UPDATE		2
UPDT: INTRODUCTION		1
		1

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 tran- !
! ! ! ! sactions are not in the same lan- !
! ! ! ! guage 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.

STANDARD PROCEDURES	PAGE	18
UPDT: DATABASE UPDATE		2
UPDT: UPDATE RULES - RESULTS		1
		2

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).

CHECKPOINT REQUEST

This facility allows you to request synchronization points during a batch update (UPDT procedure) or during a database restoration (REST or RESY procedures).
In case of ABEND, a ROLLBACK is performed, thus securing a coherent database.

Therefore, it is always possible, after an abnormal ending of the UPDT procedure, to restart the procedure without executing a restoration. However, it is recommended to delete transactions already taken into account.

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

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

INPUT OF THE CHECKPOINT FREQUENCY RATE FOR A BATCH UPDATE

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

POSITION	LENGTH	VALUE	MEANING
2	1	Y	LINE CODE
4	4	nnnn	CHECKPOINT FREQUENCY RATE
			(DEFAULT VALUE=0000)

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

CONCURRENT BATCH AND ON-LINE UPDATES

Checkpoints in the PACA15 program of the batch UPDT procedure enable it to function at the same time as TP8. Concurrent UPDT-TP8 updating should only be used with small batches of select transactions.

Running a UPDT during a TP8 session can cause a blocking of pages between two successive retrieval points, causing a slow down in on-line response time.

Two options are offered for executing the UPDT procedure including checkpoints:

- Execution with File Checkpoint.
- Execution with Program Checkpoint.

Execution with File Checkpoint

This option does not permit concurrent access. The retrieval points can only be made on random files. In case of a non-blocking abort (journal file full), the procedure can be restarted after eliminating the transactions already processed in the MBUPDT file.

If the procedure is aborted by a fatal lock due to a conflict of access between two processes, this option does not permit its automatic retrieval.

This option is selected by initializing the JCL parameters with the following values:

LEC=(R/C) ECR=(W/C) SET=SET

Execution with Program Checkpoint

This option permits concurrence between two processes: BATCH-BATCH or BATCH-TP8. Retrieval points can be made on the random files as well as on the PACA15 program via the QX file.

This option offers the same possibilities as the previous option, but it also permits automatic retrievals in case of conflict with another process. The execution of UPDT may take little longer with this option, since the system must make a process image of the PACA15 program at each retrieval point.

This option is selected by initializing the JCL parameters with the following values:

LEC=(R/C) ECR=(W/C) SET=NOTE

STANDARD PROCEDURES
UPDT: DATABASE UPDATE
UPDT: DESCRIPTION OF STEPS

PAGE

21

2
1
3

2.1.3. UPDT: DESCRIPTION OF STEPS

UPDT: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

DATABASE CONSISTENCY CHECK: PTUBAS

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Output report
-Validity report (Length=079)
SYSOUT DS

.Return code(s):
-Switch-20
1 - The Database is invalid.

TRANSACTION FORMATTING: PACA05

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index File
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Input transaction file:
-Update transactions
File MB

.Output files:
-Formatted transactions
File (FLR 167, CISZ 9413) MV
(must have capacity to contain all transactions in their
complete state, plus the elementary delete transactions
generated by the multiple delete transactions)
-Work file
SYSOUT MW

DATABASE UPDATE: PACA15

.Permanent update files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Journal file
PRMFL : \$UMCB/\$BASE.AJ AJ

.Permanent input files:
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-DSMS file of VA Pac elements
PRMFL : \$UMCBD/\$BASD.DC \$UMCBD/\$BASD.CD DC, CD
(DSM variant only)

.Input transaction file:
-Update transactions
File MV

STANDARD PROCEDURES

UPDT: DATABASE UPDATE

2

UPDT: DESCRIPTION OF STEPS

1

3

.Output report(s):

-Update report

File IE

-Erroneous-transaction list

File IF

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

.Return codes:

Switch-30

0: without error

1: with error

PREPARATION FOR PRINTING: UTI120

.Input file:

-Update report

File QC

.Output file:

-Update report

File QD

BCD PRINTING: PBCD

This printout is performed by a CONVER.

ASCII PRINTING: PASCII

This printout is performed by a CONVER.

STANDARD PROCEDURES

UPDT: DATABASE UPDATE

UPDT: EXECUTION JCL

2

1

4

2.1.4. UPDT: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.UPDT
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * BATCH UPDATE *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.UPDT *
$ NOTE * * *
$ NOTE * INPUT SYNTAX *
$ NOTE * PACBASE LINE *
$ NOTE * * *
$ NOTE * WRITE WITH TP8 *
$ NOTE * LEC=(R/C) ECR=(W/C) *
$ NOTE * SET=SET --> CHECKPOINTS FILE *
$ NOTE * SET=NOTE --> CHECKPOINTS PROGRAM *
$ NOTE * CHECKPOINT FREQUENCY IS DEFINED WITH A Y LINE *
$ NOTE * IN THE USER INPUT *
$ NOTE * (FIRST INPUT LINE) *
$ NOTE * COL 2 --> Y COL 4 --> 9999 *
$ NOTE * 9999 = CHECKPOINT FREQUENCY *
$ NOTE * * *
$ NOTE * WRITE WITHOUT TP8 *
$ NOTE * LEC=Q ECR=L *
$ NOTE * * *
$ NOTE * PRINTING IN BCD FORMAT *
$ NOTE * IMP=BCD *
$ NOTE * * *
$ NOTE * PRINTING IN ASCII FORMAT *
$ NOTE * IMP=ASCII *
$ NOTE * * *
$ NOTE *****
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ GLOBAL MBFILE=($MB.UPDT)
$ GLOBAL LEC=(R/C)
$ GLOBAL ECR=(W/C)
$ GLOBAL SET=NOTE
$ &SET 18
$ PTUBAS.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTUBAS
$ EXECUTE DUMP
$ LIMITS ,60K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,&LEC,R,$UMCB/$BASE.AR
$ PRMFL BR,&LEC,R,$UMCB/$BASE.BR
$ PRMFL AE,&LEC,R,$UMCB/$BASE.AE
$ PRMFL XE,&LEC,R,$UMCB/$BASE.XE
$ SYSOUT EI,ORG
$ SYSOUT DS,ORG
$ IF 20,ERROR
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ FILE BM,C1S,1R
$ PACA05.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PACA05
$ EXECUTE DUMP
$ LIMITS ,70K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB

```

STANDARD PROCEDURES

UPDT: DATABASE UPDATE

2

UPDT: EXECUTION JCL

1

4

```

$      PRMFL  LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$      PRMFL  AE,&LEC,R,$UMCB/$BASE.AE
$      PRMFL  XE,&LEC,R,$UMCB/$BASE.XE
$      PRMFL  AN,&LEC,R,$UMCB/$BASE.AN
$      PRMFL  BN,&LEC,R,$UMCB/$BASE.BN
$      PRMFL  AR,&LEC,R,$UMCB/$BASE.AR
$      PRMFL  BR,&LEC,R,$UMCB/$BASE.BR
$      SYSOUT EI,ORG
$      FILE   MV,M1S,10R
$      FILE   MW,,20R
$      FILE   MB,C1R
$      IF     20,ERROR
$ PACA15.
$      OPTION CBL74
$      LIBRARY LA,LB
$      SELECT $UMCS/$OBJBT.PACA15
$      EXECUTE DUMP
$      LIMITS 100,243K,,50K
$      PRMFL  1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL  LA,R/C,R,$UMCS/$FILS.OBJLIB
$      PRMFL  LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$      PRMFL  AE,&LEC,R,$UMCB/$BASE.AE
$      PRMFL  XE,&LEC,R,$UMCB/$BASE.XE
$      PRMFL  DC,&LEC,R,$UMCBD/$BASD.DC
$      PRMFL  CD,&LEC,R,$UMCBD/$BASD.CD
$      PRMFL  AN,&ECR,R,$UMCB/$BASE.AN
$      PRMFL  BN,&ECR,R,$UMCB/$BASE.BN
$      PRMFL  AR,&ECR,R,$UMCB/$BASE.AR
$      PRMFL  BR,&ECR,R,$UMCB/$BASE.BR
$      PRMFL  AJ,&ECR,R,$UMCB/$BASE.AJ
$      FILE   MV,M1
$      FILE   QX,,500R
$      SYSOUT EI,ORG
$      FILE   IE,E1S,100L
$      FILE   IF,F1S,10L
$      DATA  .U
FILE   FC/AN/,NBUF/8/,BFSZ/4096/
FILE   FC/BN/,NBUF/8/,BFSZ/4096/
FILE   FC/AR/,NBUF/8/,BFSZ/4096/
FILE   FC/BR/,NBUF/8/,BFSZ/4096/
$ UTI120.
$      OPTION CBL74
$      SELECT $UMCS/$OBJBT.UTI120
$      EXECUTE DUMP
$      FILE   QC,E1R
$      FILE   QD,E2S,100L
$      GOTO   P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$      CONVER
$      LIMITS ,,50K
$      FILE   IN,E2R
$      FILE   ",F1R
$      SYSOUT OT,&RMTB
$      OUTPUT GBCD,MEDIA/3
$      GOTO   END
$ PASCII.
$ ASCII-PRINT 132 CH.
$      CONVER
$      LIMITS ,,50K
$      FILE   IN,E2R
$      FILE   ",F1R
$      SYSOUT OT,&RMTA
$      OUTPUT ASCII,MEDIA/7
$ END.
$      IF     20,ERROR
$      CONVER
$      DATA  IN
***** UPDT - NORMAL END OF RUN *****
$      SYSOUT OT,ORG
$      OUTPUT MEDIA/03
$ ERROR.
$      ENDJOB

```


STANDARD PROCEDURES	PAGE	25
UPDP: DATABASE UPDATE FROM PAF TABLES		2
UPDP: INTRODUCTION		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.

STANDARD PROCEDURES	
UPDP: DATABASE UPDATE FROM PAF TABLES	
UPDP: INPUT - PROCESSING - RESULTS	

2
2
2

2.2.2. UPDP: INPUT - PROCESSING - RESULTS

UPDP: INPUT-PROCESSING-RESULTS

USER INPUT

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

```

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

```

UPDATE RULES

Update transactions are not sorted.

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

```

-----
! Pos. ! Length ! Value ! Meaning !
-----
! 2 ! 10 ! 'ASSIGN' ! Table code !
! 12 ! 8 ! uuuuuuuu ! User code !
! 20 ! 8 ! pppppppp ! Password !
! 28 ! 3 ! bbb ! Library code !
! 31 ! 4 ! ssss ! Session number !
! ! ! ' ' ! current session !
! 35 ! 1 ! 'T' ! Session status: Test session !
! 36 ! 3 ! nnn ! No line numbering !
! 39 ! 1 ! 'A' or ! Language code, useful if the !
! ! ! 'F' ! 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 !
-----

```

STANDARD PROCEDURES

UPDP: DATABASE UPDATE FROM PAF TABLES

2

UPDP: INPUT - PROCESSING - RESULTS

2

2

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 proces-
!    !       !         ! sed between two pauses or
!    !       !         ! checkpoints
! 16  !       4 ! 'UPDT'  ! Update procedure
!    !       !         !
! 20  !       2 ! nn      ! OS/2, UNIX, WINDOWS NT:
!    !       !         ! Pause time, in seconds, bet-
!    !       !         ! ween 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
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Output report
-Validity report (Length=079)
SYSOUT DS

.Return code(s):
-Switch-20
1 - The Database is invalid.

TRANSACTION FORMATTING: PAF900

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index File
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Input transaction file:
-Update transactions
PRMFL : \$UMCU/\$MV.UPDP GY

.Output files:
-Formatted transactions
File (FLR 167, CISZ 9413) MV
(must have capacity to contain all transactions in their
complete state, plus the elementary delete transactions
generated by the multiple delete transactions)
-Work file
SYSOUT MW

DATABASE UPDATE: PACA15

.Permanent update files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Journal file
PRMFL : \$UMCB/\$BASE.AJ AJ

.Permanent input files:
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-DSMS file of VA Pac elements
PRMFL : \$UMCBD/\$BASD.DC \$UMCBD/\$BASD.CD DC, CD
(DSM variant only)

.Input transaction file:
-Update transactions
File MV

STANDARD PROCEDURES	PAGE	29
UPDP: DATABASE UPDATE FROM PAF TABLES		2
UPDP: DESCRIPTION OF STEPS		3

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

.Return codes:
 Switch-30
 0: without error
 1: with error

PREPARATION FOR PRINTING: UTI120

.Input file:
 -Update report
 File QC

.Output file:
 -Update report
 File QD

BCD PRINTING: PBCD

This printout is performed by a CONVER.

ASCII PRINTING: PASCII

This printout is performed by a CONVER.

STANDARD PROCEDURES

UPDP: DATABASE UPDATE FROM PAF TABLES

UPDP: EXECUTION JCL

2

2

4

2.2.4. UPDP: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.UPDP
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * BATCH UPDATE FROM PAF TABLES *
$ NOTE * * *
$ NOTE * THE FILE OF INPUT TRANSACTIONS IS PRODUCED *
$ NOTE * BY A PAF EXTRACTOR PROGRAM *
$ NOTE * $UMCU/$MV.UPDP *
$ NOTE * * *
$ NOTE * WRITE WITH TP8 *
$ NOTE * LEC=(R/C) ECR=(W/C) *
$ NOTE * SET=SET --> CHECKPOINTS FILE *
$ NOTE * SET=NOTE --> CHECKPOINTS PROGRAM *
$ NOTE * CHECKPOINT FREQUENCY IS DEFINED WITH A CHECKP *
$ NOTE * LINE IN THE USER INPUT *
$ NOTE * (FIRST INPUT LINE) *
$ NOTE * * *
$ NOTE * WRITE WITHOUT TP8 *
$ NOTE * LEC=Q ECR=L *
$ NOTE * * *
$ NOTE * PRINTING IN BCD FORMAT *
$ NOTE * IMP=BCD *
$ NOTE * * *
$ NOTE * PRINTING IN ASCII FORMAT *
$ NOTE * IMP=ASCII *
$ NOTE * * *
$ NOTE *****
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ GLOBAL MBFILE=($MV.UPDP)
$ GLOBAL LEC=(R/C)
$ GLOBAL ECR=(W/C)
$ GLOBAL SET=NOTE
$ &SET 18
$ PTUBAS.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTUBAS
$ EXECUTE DUMP
$ LIMITS ,60K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,&LEC,R,$UMCB/$BASE.AR
$ PRMFL BR,&LEC,R,$UMCB/$BASE.BR
$ PRMFL AE,&LEC,R,$UMCB/$BASE.AE
$ PRMFL XE,&LEC,R,$UMCB/$BASE.XE
$ SYSOUT EI,ORG
$ SYSOUT DS,ORG
$ IF 20,ERROR
$ PAF900.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PAF900
$ EXECUTE DUMP
$ LIMITS ,70K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,&LEC,R,$UMCB/$BASE.AE
$ PRMFL XE,&LEC,R,$UMCB/$BASE.XE
$ PRMFL AN,&LEC,R,$UMCB/$BASE.AN
$ PRMFL BN,&LEC,R,$UMCB/$BASE.BN
$ PRMFL AR,&LEC,R,$UMCB/$BASE.AR
$ PRMFL BR,&LEC,R,$UMCB/$BASE.BR
$ PRMFL GY,R,R,$UMCU/&MBFILE

```

STANDARD PROCEDURES

UPDP: DATABASE UPDATE FROM PAF TABLES

2

2

UPDP: EXECUTION JCL

4

```

$      SYSOUT  EI,ORG
$      FILE    MV,M1S,10R
$      FILE    MW,,20R
$      IF      20,ERROR
$ PACA15.
$      OPTION  CBL74
$      LIBRARY LA,LB
$      SELECT  $UMCS/$OBJBT.PACA15
$      EXECUTE DUMP
$      LIMITS  100,243K,,50K
$      PRMFL   1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL   LA,R/C,R,$UMCS/$FILS.OBJLIB
$      PRMFL   LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$      PRMFL   AE,&LEC,R,$UMCB/$BASE.AE
$      PRMFL   XE,&LEC,R,$UMCB/$BASE.XE
$      PRMFL   DC,&LEC,R,$UMCBD/$BASD.DC
$      PRMFL   CD,&LEC,R,$UMCBD/$BASD.CD
$      PRMFL   AN,&ECR,R,$UMCB/$BASE.AN
$      PRMFL   BN,&ECR,R,$UMCB/$BASE.BN
$      PRMFL   AR,&ECR,R,$UMCB/$BASE.AR
$      PRMFL   BR,&ECR,R,$UMCB/$BASE.BR
$      PRMFL   AJ,&ECR,R,$UMCB/$BASE.AJ
$      FILE    MV,M1
$      FILE    QX,,500R
$      SYSOUT  EI,ORG
$      FILE    IE,E1S,100L
$      FILE    IF,F1S,10L
$      DATA   .U
FILE    FC/AN/,NBUF/8/,BFSZ/4096/
FILE    FC/BN/,NBUF/8/,BFSZ/4096/
FILE    FC/AR/,NBUF/8/,BFSZ/4096/
FILE    FC/BR/,NBUF/8/,BFSZ/4096/
$ UTI120.
$      OPTION  CBL74
$      SELECT  $UMCS/$OBJBT.UTI120
$      EXECUTE DUMP
$      FILE    QC,E1R
$      FILE    QD,E2S,100L
$      GOTO    P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$      CONVER
$      LIMITS  ,,50K
$      FILE    IN,E2R
$      FILE    ",F1R
$      SYSOUT  OT,&RMTB
$      OUTPUT  GBCD,MEDIA/3
$      GOTO    END
$ PASCII.
$ ASCII-PRINT 132 CH.
$      CONVER
$      LIMITS  ,,50K
$      FILE    IN,E2R
$      FILE    ",F1R
$      SYSOUT  OT,&RMTA
$      OUTPUT  ASCII,MEDIA/7
$ END.
$      IF      20,ERROR
$      CONVER
$      DATA   IN
***** UPDP - NORMAL END OF RUN *****
$      SYSOUT  OT,ORG
$      OUTPUT  MEDIA/03
$ ERROR.
$      ENDJOB

```

STANDARD PROCEDURES	PAGE	32
GPRT: GENERATION AND PRINTING		2
GPRT: INTRODUCTION		3
		1

2.3. GPRT: GENERATION AND PRINTING

2.3.1. GPRT: INTRODUCTION

GENERATION-PRINT (GPRT)

This procedure ensures the printing and generation of the documentation, when requested by the user, of any entity that can be generated.

It has the following characteristics:

14 monitors make up the procedure (PACBA, PACBE, ...). Each monitor runs a sequence of sub-routines associated with a type of generation.

Communication between a monitor and its sub-routines is ensured by the area called LINKAGE-SECTION. Communication between monitors is ensured by a work file.

Since user requests are often diverse, each monitor processes, in an integrated manner, the generation and the preparation of the printing requests for the category it manages. Each category is identified by one character code as follows:

- A : Specifications Dictionary
- B : Database blocks (DBD)
- D : Data
- E : Screens (OSD)
- G : Client part for Client/Server
- K : Error Messages for Client/Server
- L : Error messages and revamping
- M : User manuals
- N : Volumes (PDM)
- P : Batch programs (BSD)
- Q : Relational-SQL Database blocks
- R : Reverse Eng. Programs (REV)
- V : Server part for Client/Server

This code is referenced again in the names given to the monitors, sub-routines, files and procedure reports. For programs, the code is the fourth character of the FILE-CODE. Examples:

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

	PAGE	33
STANDARD PROCEDURES		2
GPRT: GENERATION AND PRINTING		3
GPRT: INTRODUCTION		1

For files or reports, the code is the last character of the FILE-CODE. Examples:

IA: General printing of command chain.

GP: Generated file of batch programs.

. Following the execution of the first monitor that recognizes user requests and saves them (if necessary) in the production environment, the rest of the monitors are activated, if appropriate, in the following order:

- Screens
- Batch programs and GIP interface
- Volume
- Release 7 error messages and Revamping
- Client for Client/Server
- Server for Client/Server
- Client/Server error messages
- Database Blocks
- SQL Database
- COBOL programs
- Specifications Dictionary, DATAs

. Each monitor is structured in the same manner:

- 'Extraction' programs (3x),
- 'Preparation' programs (4x),
- 'Generation' programs (8x),
- 'Formatting' programs (92),
- 'Printing' programs (90).

These codes are found in the last two characters of the sub-programs codes of the procedure. Examples:

- PACB40: preparation of database blocks.

- PACE80: screen generation.

. In addition, special codes have been adopted for the FILE-CODES of the files. They represent the use of the procedure files:

- G : Generations
- I : Reports
- J : Printing requests
- K : Printing preparation
- L : Error messages

	PAGE	34
STANDARD PROCEDURES		2
GPRT: GENERATION AND PRINTING		3
GPRT: INTRODUCTION		1

M : Transactions
 Q : Skeletons
 W : Work

They are found in the first character of the FILE-CODES of the procedure files.
 Examples:

GL : Generated error messages.

IM : Printing of user manuals.

IN : Printing of volumes.

. The user error message file (GL), 7.0 type, is updated from the last generated version of this file (LG).

The installation procedure includes the following codes for these two files:

\$UMCU/\$FILG/LG&USER
 \$UMCU/\$FILG/GL&USER

After every generation, a shift on these files is performed by the GCOS8 FILSYS utility.

After this shift, the file to be integrated into the user applications using EMLD, EMUP or a user procedure always corresponds to the following string:

\$UMCU/\$FILG/LG&USER

This file is sorted according to the ASCII collating sequence.

. The user error message file (GK), Client/Server type, is updated from the last generated version of this file (LK).

The installation procedure includes the following codes for these two files :

\$UMCU/\$FILG/LK&USER (LK)
 \$UMCU/\$FILG/GK&USER (GK)

	PAGE	35
STANDARD PROCEDURES		
GPRT: GENERATION AND PRINTING		2
GPRT: INTRODUCTION		3
		1

After every generation, a shift on these files is performed by the GCOS8 FILSYS utility.

After this shift, the file to be integrated into the user applications using EMLD, EMUP or a user procedure always corresponds to the following string:

\$UMCU/\$FILG/LK&USER

This file is sorted according to the EBCDIC collating sequence.

Ordinary volume print is directed to the IN FILE-CODE file. This documentation can also be edited on a GN FILE-CODE file with the skip character ASA in first position of each record (length = 265) for special print processes.

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 3-character code.

The first character identifies the nature:

- . 'L': List of occurrences
- . 'D': Description of occurrences, including Definition, Description and General Documentation
- . 'G': Generation of source code for the selected occurrence
- . 'P': Print Volume occurrence. The second character must be 'C', and the third '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.

STANDARD PROCEDURES	PAGE	37
GPRT: GENERATION AND PRINTING		2
GPRT: STRUCTURE OF REQUESTS		3
		2

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	Model entity: Functional Integrity Constraint
MO	Model entity: Object
MP	Model entity: Property
MR	Model entity: Relationship
O	Dialog, Screen, C/S Screen, Business Component Folder, Folder View
P	Program
Q	User-Defined Relationship
R	Report
S	Segment
T	Text
V	Volume
\$	User Entity Occurrence

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.

	PAGE	38
STANDARD PROCEDURES		
GPRT: GENERATION AND PRINTING		2
GPRT: STRUCTURE OF REQUESTS		3
		2

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 selection field in which the user can specify whether the selection is to be made:

- . On the whole set of keywords (SPACE),
- . On the keywords automatically derived from the name (L),
- . On explicit keywords (M).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	1		ACTION CODE
2	2	blank A - 99 90 91 92 94 95 96	<p>PROCESSING SEQUENCE ORDER</p> <p>1. LIST/DESCRIPTION PRINT REQUESTS:</p> <p>This field is used to specify the sequence in which your Print Requests will be processed and printed.</p> <p>The Request is processed according to its position in the input sequence displayed in the CH: GP screen.</p> <p>The List/Description Print Request is processed according to the order number entered in this field.</p> <p>2. GENERATION REQUESTS:</p> <p>Generation Requests (including Volumes) are automatically sorted by entity types:</p> <p>Programs Screens Database Blocks Error messages Data structures Volumes (PDM)</p> <p>NOTE: Any modification of these values is ignored; no error message is issued.</p>
3	4		<p>GENERATION-PRINT COMMAND</p> <p>NOTE: Input of the entity code is required or optional depending on the command. The following indicators describe the various options:</p> <p>(A) Required occurrence code input (Batch column 9).</p> <p>(B) Optional occurrence code input. If omitted, all occurrences of the entity type are listed in the user's hierarchical view.</p> <p>(C) Occurrence code input not allowed. All occurrences of the entity type are listed in the user's hierarchical view.</p> <p>(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; you will find below what options are possible. This corresponds to batch columns 31 to 80 incl.</p> <p>NOTE: Each command has different requirements with</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			<p>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:</p> <p>(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.</p> <p>(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.</p> <p>(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.</p> <p>(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.</p> <p>(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.</p> <p>(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.</p> <p>(7) PRINT VOLUME BY CHAP/SUBCHAP AND CODE: _ _ _ Specify the chapter and/or subchapter. Enter 'C' for chapter followed by the chapter code, or 'S' for subchapter followed by the chapter and subchapter codes. In batch mode use columns 23 through 27.</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			<p>(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.</p> <p>THESAURUS -----</p>
		DCK	<p>(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.</p>
		LCK	<p>(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.</p>
			<p>TEXTS -----</p>
		DCT	<p>(A) Description of selected Text. NOTE: If you enter an "*" in the ENTITY CODE field, descriptions of all Text occurrences will be printed, sorted by code.</p>
		DTT	<p>(B) (6) Descriptions of Text occurrences, sorted by type.</p>
		L*T	<p>List of Texts with their paragraphs titles, sorted by code.</p>
		LCT	<p>(C) List of Text occurrences, sorted by code.</p>
		LKT	<p>(2) List of Text occurrences whose names and/or explicit Keywords contain the Keyword(s) specified.</p>
		LTT	<p>(6) List of Text occurrences, sorted by type.</p>
			<p>VOLUMES -----</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		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 keyword(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'.
			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.

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		LKS	(C) (2) List of Segments whose names and/or explicit keywords contain the keyword(s) specified.
		DCS	(B) (D: when entity code has been entered) (3) NOTE: Enter the Data Structure code in the ENTITY CODE field, and the Segment code(s) on the continuation line(s). A complete description of the Segment(s). This includes cross-references to Programs and Screens for the Data Structure and to all entities for the Segment(s) and a list of associated Reports and Segments. For Segments defined as tables with the Pac-tables function, a list of sub-schemas and sub-systems is printed. NOTE: To get the associated text for both the Segment and the Data Structure, use print option "2". INPUT AIDS -----
		DCI	(C) A complete description of the input aid(s) including a list of uses of the input aid(s) in other entities. The information is sequenced by PIA code.
		LCI	(C) A list of input aids sequenced by the PIA code.
		LKI	(C) (2) A list of the input aids whose names and/or explicit keywords contain the keyword(s) specified.
		LXI	(C) List of all Cross-References (PIA Calls) as defined on the PIA description screen sequenced by the value of this field.
			DATABASE BLOCKS -----
		DTB	(B) (6) Description(s) of database blocks of the type specified including cross-references to other blocks and screens. Note: To get the associated text, use print option "2"
		FLB	(C) (D) (4) (8)

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

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

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

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		LKM	(C) (2) A list of the Method entities whose names and/or explicit keywords contain the keyword(s) specified.
			USER ENTITIES -----
		DCF	(B) A complete description of the User Entity(s). The information is sequenced by User Entity code.
		DCQ	(B) A complete description of the User-Defined Relationship. The information is sequenced by Relationship code.
		DC\$	(B) A complete description of the User Entity Occurrence(s). The information is sequenced by user entity type code.
		LCF	(C) List of User Entities sequenced by code.
		LCQ	(C) List of User-Defined Relationships sequenced by code.
		LC\$	(C) List of User Entity Occurrences sequenced by User Entity type code.
		LK\$	(2) (A) A list of the User Entity Occurrences whose names and/or explicit keywords contain the keyword(s) specified.
		LKF	(2) (C) A list of the User Entities whose names and/or explicit keywords contain the keyword(s) specified.
		LKQ	(2) (C) A list of the User-Defined Relationships whose names and/or explicit keywords contain the keyword(s) specified.
			NOTE ----
			For all printing by keyword, you can specify the type of selection (BLANK, L or M) on the print line. Keywords are indicated on the continuation line sent back by VisualAge Pacbase.

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

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

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

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			<p>Selected library and higher level libraries. In case of duplicates, the lines from the lower level library are taken into account.</p> <p>NOTE: IN GENERATION THE VALUE 'C' IS AUTOMATICALLY ASSIGNED BY THE SYSTEM.</p> <p>I Selected library and lower and higher level libraries.</p> <p>U Selected library only.</p> <p>A Selected library and higher level libraries with display of duplicates.</p> <p>> Higher level libraries only.</p> <p>< Lower level libraries only.</p> <p>Z Selected library and lower level libraries.</p>
6	1		<p>PRINT OPTION</p> <p>This field does not appear on the "C4" screen format option.</p> <p>Used to indicate that sub-reports be included.</p> <p>1 Default</p> <p>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.</p>
7	1		<p>VALIDATION OF COMMAND REQUEST</p> <p>This field does not appear on the "C2" screen format option.</p> <p>blank The value in the COMMAND FOR PRINT REQUEST field is not to be taken into account.</p> <p>V The COMMAND FOR PRINT REQUEST is validated.</p> <p>NOTE: These commands must be re-validated each time a request is made.</p>
8	1		<p>CONTINUATION OF REQUEST INDICATOR</p> <p>blank No continuation line is requested.</p> <p>* A continuation line is requested (or displayed) for this GP command.</p> <p>For some specific Generation-Print Requests, this field is automatically filled by VA Pac (for instance a Reques by Keywords). You must then fill in the continuation line's input fields.</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			NOTE: A maximum of 5 continuation lines is authorized.
9	50		<p>COMMAND LABEL / SYSTEM RESPONSE</p> <p>This field has three functions:</p> <ul style="list-style-type: none"> - 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		<p>CONTINUATION LINE</p> <p>This line is displayed on-line. It represents columns 31 through 80 on Batch Form 'Z'.</p> <p>This line serves many purposes, among them:</p> <ul style="list-style-type: none"> . 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.
11	3	blank JOB SUB	<p>JOB SUBMISSION REQUEST</p> <p>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.</p> <p>No job submission. Update the AG file.</p> <p>Job submission.</p> <p>NOTE: For IMS, system messages are displayed. See USER'S MANUAL, chapter "CHOICE: ACCESS COMMANDS", subchapter "SPECIAL CHOICES: IMS VERSION".</p> <p>Job submission.</p> <p>NOTE: For IMS, system messages are not displayed.</p>
			<p>SYSTEM RESPONSE REQUEST</p> <p>The following fields appear in the COMMAND LABEL/ SYS-</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			TEM 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	blank or C E I R S	<p>TYPE TO SELECT</p> <p>A. TYPE TO SELECT (2-character field): Used to specify an occurrence type when requesting a List or Description sorted by type.</p> <p>B. FORMAT TO SELECT (1-character field): Used to specify the Segment format when entering a DCS command.</p> <p>Printing of data related to validations and updates performed by user programs on the Segment's Data Elements. In addition, internal and input formats are printed.</p> <p>E Input format only.</p> <p>I Internal format only.</p> <p>R Validations, updates, relational names.</p> <p>S Output format only.</p>
13	1	1 0	<p>CARDS IN FRONT / UPPERCASE SHIFT</p> <p>GENERATION -----</p> <p>Enter the one-character code that identifies the job card to be inserted before the generated occurrence. Default: Code entered in the Library Definition.</p> <p>NOTE: This value may be overridden on the occurrence's Definition.</p> <p>Also see Subchapter "OPTIONAL CONTROL CARDS UPDATING", Chapter "DATABASE MANAGEMENT", OPTION CODE field in the VA Pac TUI User Interface Guide (Ref. DD USE).</p> <p>SHIFT TO UPPERCASE FOR VOLUMES -----</p> <p>Volumes ('V' entity) are printed in uppercase characters with the UPC command.</p> <p>1 YES. 0 NO (Default option).</p>
14	1		<p>CARDS IN FRONT MAP/UPPERCASE SHIFT</p> <p>SCREEN GENERATION -----</p> <p>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.</p>

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			<p>Also see: OPTION CODE and INPUT PARAMETERS fields in the "OPTIONAL CONTROL CARDS UPDATING" Subchapter, "DATABASE MANAGEMENT" Chapter in the VA Pac TUI User Interface Guide (Ref. DD USE).</p> <p>\$</p> <p>No generation of map. (Use this value in conjunction with the CONTROL CARDS IN BACK OF MAP field.)</p> <p>SHIFT TO UPPERCASE FOR LIST/DESCRIPTION PRINT OUTPUT -----</p> <p>Print output shifted to uppercase with UPC command.</p> <p>1 0</p> <p>YES. NO (Default option).</p>
15	1		<p>CARDS IN BACK / UPPERCASE SHIFT</p> <p>GENERATION -----</p> <p>Enter the one-character code that identifies the job card to be inserted after the generated occurrence.</p> <p>Default: Code entered on the Library Definition.</p> <p>NOTE: This value may be overridden on the occurrence Definition.</p> <p>SHIFT TO UPPERCASE FOR PRINTED ERROR MESSAGES -----</p> <p>Error messages are printed in uppercase characters with the UPC command.</p> <p>1 0</p> <p>YES. NO (Default option).</p>
16	1		<p>CONTROL CARDS AFTER MAP</p> <p>The one-character code that identifies the job card to be inserted after each generated screen map.</p> <p>\$</p> <p>No generation of map.</p>
17	1	blank L M	<p>SELECTION OF KEYWORD TYPE</p> <p>Selection on both implicit and explicit keywords.</p> <p>Selection on implicit keywords only.</p> <p>Selection on explicit keywords only.</p>
18	1		<p>REQUEST FOR SELECTIVE VOLUME PRINT</p> <p>Field displayed with PCV command only.</p> <p>NOTE: Applicable only when the Volume has a Chapter/ Subchapter Description Organization Mode</p>

STANDARD PROCEDURES

PAGE

59

GPRT: GENERATION AND PRINTING

2

GPRT: GENERATION/PRINTING COMMANDS

3

3

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE (Value "0" in corresponding field in Volume Definition).
		blank C S	Print the whole Volume (default value) Print the selected chapter (see next field) Print the selected subchapter (see next two fields)
19	2		CODE OF THE CHAPTER TO BE PRINTED Field displayed with PCV command only. Code of the chapter to be printed, or the chapter that contains the subchapter to be printed.
20	2		CODE OF THE SUBCHAPTER TO BE PRINTED Field displayed with PCV command only. Code of the subchapter to be printed.
21	8		CODE OF RECIPIENT USER FOR JCL COPY This field is reserved for on-line use. If you have a 4-level authorization, this field allows you to initialize another user's JCL lines. To do so, when the JCL lines are displayed, override your user code with that of the other user. Press the ENTER key.

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.

STANDARD PROCEDURES	
GPRT: GENERATION AND PRINTING	
GPRT: USER INPUT AND RESULTS	

PAGE	61
	2
	3
	4

PARAMETERS IN USE

&USER: Several generations can be executed simultaneously if the names of the generated files are parameterized by a user code. Therefore, any new user needs to create the following files:

1. CORRESPONDING GENERATION FILES

-OLSD	\$UMCU/\$FILG.GE&USER
-BATCH SD (GIP interface)	\$UMCU/\$FILG.GP&USER \$UMCU/\$FILG.GI&USER
-Volumes (manuals)	\$UMCU/\$FILG.GN&USER
-Volumes RTF	\$UMCU/\$FILG.G6&USER
-Rel. 7 Error Mess.	\$UMCU/\$FILG.GL&USER \$UMCU/\$FILG.LG&USER
-Local revamping	\$UMCU/\$FILG.GT&USER
-Client for C/S	\$UMCU/\$FILG.GG&USER
-Server for C/S	\$UMCU/\$FILG.GV&USER
-C/S Error Messages	\$UMCU/\$FILG.GK&USER \$UMCU/\$FILG.LK&USER
-Databases	\$UMCU/\$FILG.GB&USER
-SQL Databases	\$UMCU/\$FILG.GQ&USER
-Reversed programs	\$UMCU/\$FILG.GR&USER
-DATA Generation	\$UMCU/\$FILG.GD&USER

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
GPRT: USER INPUT AND RESULTS

PAGE

62

2
3
4

2. REQUESTS FILE FOR USING GPRT ON TSS

\$UMCU/\$MB.GP&USER

3. CYCLE SHIFT PROCEDURE FOR RELEASE 7 ERROR MESSAGE FILE

\$UMCU/\$JCL.R&USER

3. CYCLE SHIFT PROCEDURE FOR C/S ERROR MESSAGE FILE

\$UMCU/\$JCL.S&USER

Depending on whether the procedure is submitted in batch or in on-line mode (JOB function), the ME file must be assigned a FILE or a PRMFL.

&FILE: NOTE if GPRT is submitted in batch mode.
FILE if GPRT is submitted on-line.

&PRMFL: PRMFL if GPRT is submitted in batch mode.
NOTE if GPRT is submitted on-line.

&ACCES: W if '+AG' request.
R otherwise.

STANDARD PROCEDURES	PAGE	63
GPRT: GENERATION AND PRINTING		2
GPRT: DESCRIPTION OF STEPS		3
		5

2.3.5. GPRT: DESCRIPTION OF STEPS

GPRT : DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

GENERATION-PRINT: PACBA,....PACBED

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

- .PACBA : Request processing
(PACA10, PACA20, PACR20)
- .PACBE : Screen generation
(PACE30, PACE80)
- .PACBP : Batch program generation
(PACP30, PACP40, PACP80, PACP92)
- .PACBN : Volume generation-print
(PACN30, PACNT3, PACN40, PACN50, PACN80)
- .PACBL : V7 error message generation-print
and Application Revamping
(PACL30, PACL40, PACL80, PACL90, PACLTA)
- .PACBG : Client generation for Client/Server
(PACG3C, PACG4C, PACG8C)
- .PACBV : Server generation for Client/Server
(PACG3S, PACG4S, PACG8S)
- .PACBK : Client/Server error message generation
(PACK30, PACK80, PACK90)
- .PACBB : Database description generation
(PACB30, PACB40, PACB80)
- .PACBQ : Relational Database generation
(PACQ30)
- .PACBR : Reverse Eng. programs generation
(PACC30, PACC40, PACC80)

	PAGE	64
STANDARD PROCEDURES		2
GPRT: GENERATION AND PRINTING		3
GPRT: DESCRIPTION OF STEPS		5

.PACBD : Data generation and extraction of general
documentation
(PACD30, PACD40, PACD80)

.PACBED: General documentation print
(PACD90)

The generated documentation depends on the generation-print requests submitted and processed. The volume of the generated documentation and of the temporary files is extremely variable. Banners at the beginning and the end of the documentation display the user code, identifying the user that submitted the print-out request.

GPRT output is retrieved from the following files:

SCREENS	: GE
PROGRAMS	: GP
MANUALS	: GN
MANUALS RTF	: G6
V7 ERROR MESSAGES	: LG
REVAMPING	: GT
CLIENTS FOR C/S	: GG
SERVERS FOR C/S	: GV
C/S ERROR MESSAGES	: LK
DATABASE BLOCKS	: GB
(HIERARCHICAL & CODASYL)	
SQL BLOCKS	: GQ
REVERSE ENG. PROGRAMS	: GR
DATA	: GD

Some programs called by the monitor can generate specific return codes which are found in the GPRT execution output chart. These codes only have a documentary value.

.PACA10 (Retrieval of transactions):

0 : OK,
2 : OK with presence of the '+AG' command,
8 : No request. In this case, the procedure terminates.

.Extractors or generators (30 or 40):

0 : OK - No generation,
4 : OK - Generation,
Other: Errors.

LIST AND CHARACTERISTICS OF FILES

.VISUALAGE PACBASE FILES:

-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR

-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN

-Error Message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

-Request file
PRMFL : \$UMCB/\$BASE.AG \$UMCB/\$BASE.XG AG, XG

-PEI files
PRMFL : \$UMCB/\$BASE.AB \$UMCB/\$BASE.XB AB, XB
PRMFL : \$UMCB/\$BASE.AC \$UMCB/\$BASE.XC AC, XC

-User Parameter file
PRMFL : \$UMCB/\$BASE.AP \$UMCB/\$BASE.XP AP, XP

.GENERATED FILES:

-Screens
PRMFL : \$UMCU/\$FILG.GE&USER GE
(GFRC-ASCII, FLR 80)

-Batch Programs
PRMFL : \$UMCU/\$FILG.GP&USER GP
(GFRC-ASCII, FLR 80)

-GIP Interface
PRMFL : \$UMCU/\$FILG.GI&USER GI
(GFRC-ASCII, FLR 80)

-Volumes
PRMFL : \$UMCU/\$FILG.GN&USER GN
(GFRC-ASCII, FLR 265)

-Volumes in RTF format
PRMFL : \$UMCU/\$FILG.G6&USER G6
(GFRC-ASCII, FLR 345)

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: DESCRIPTION OF STEPS

3

5

-Release-7 error messages PRMFL : \$UMCU/\$FILG.GL&USER (GFRC-ASCII, FLR 90)	GL
-Release-7 input error messages PRMFL : \$UMCU/\$FILG.LG&USER (GFRC-ASCII, FLR 90)	GL
-Revamping PRMFL : \$UMCU/\$FILG.GT&USER (GFRC-ASCII, FLR 180)	GT
-Clients for C/S PRMFL : \$UMCU/\$FILG.GG&USER (GFRC-ASCII, FLR 80)	GG
-Servers for C/S PRMFL : \$UMCU/\$FILG.GV&USER (GFRC-ASCII, FLR 80)	GV
-C/S error messages PRMFL : \$UMCU/\$FILG.GK&USER (GFRC-ASCII, FLR 100)	GK
-C/S input error messages PRMFL : \$UMCU/\$FILG.LK&USER (GFRC-ASCII, FLR 100)	LK
-Databases PRMFL : \$UMCU/\$FILG.GB&USER (GFRC-ASCII, FLR 80)	GB
-Relationnal Databases PRMFL : \$UMCU/\$FILG.GQ&USER (GFRC-ASCII, FLR 80)	GQ
-COBOL Programs (REVERSE ENGIN.) PRMFL : \$UMCU/\$FILG.GR&USER (GFRC-ASCII, FLR 80)	GR
-Data PRMFL : \$UMCU/\$FILG.GD&USER (GFRC-ASCII, FLR 80)	GD

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
GPRT: DESCRIPTION OF STEPS

PAGE

67

2
3
5

.OUTPUT REPORTS:

-Execution report	
SYSOUT	IA
-General Documentation	
SYSOUT	ID
-PEI	
SYSOUT	IH
-C/S error messages	
SYSOUT	IK
-Release 7 error messages	
SYSOUT	IL
-Volumes	
SYSOUT	IN
-ABORT report (I/O error)	
SYSOUT	EI

.WORK FILES:

-Screen generation error	
File (FLR 166, CISZ = 4,608)	EE
-Client generation error	
File (FLR 180, CISZ = 4,608)	EG
-Volume generation error	
File (FLR 180, CISZ = 4,608)	EN
-Program generation error	
File (FLR 180, CISZ = 4,608)	EP
-SQL Block generation error	
File (FLR 180, CISZ = 4,608)	EQ
-Server generation error	
File (FLR 180, CISZ = 4,608)	EV
-Extracted requests	
File (FLR 80, CISZ = 1,100)	JG
-Extracted database descriptions	
File (FLR 180, CISZ = 4,608)	KB
-Extracted general documentation	
File (FLR 180, CISZ = 4,608)	KD

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

GPRT: DESCRIPTION OF STEPS

PAGE

68

2
3
5

-Extracted screens File (FLR 180, CISZ = 4,608)	KE
-List of requests File (FLR 180, CISZ = 4,608)	KF
-Extracted clients File (FLR 180, CISZ = 4,608)	KG
-7.0 User Manuals or error messages File (FLR 180, CISZ = 4,608)	KM
-Volumes File (FLR 180, CISZ = 4,608)	KN
-Extracted programs File (FLR 166, CISZ = 2,048)	KP
-Extracted SQL databases FILE (FLR 180, CISZ = 4,608)	KQ
-Lists by keywords FILE (FLR 180, CISZ = 4,608)	KS
-List of requests with titles FILE (FLR 180, CISZ = 4,608)	KU
-Extracted servers FILE (FLR 180, CISZ = 4,608)	KV
-Error Messages to print FILE (FLR 180, CISZ = 4,608)	LI
-Entered requests FILE (FLR 80, CISZ = 512)	ME
-Processed requests FILE (FLR 150, CISZ = 1,240)	MG
-Inter-Monitor transfer FILE (FLR 80, CISZ = 848)	YL
-GPRT report printing FILE (FLR 134, CISZ = 1,388)	YM

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
GPRT: DESCRIPTION OF STEPS

PAGE

69

2
3
5

.VISUALAGE PACBASE SYSTEM FILES:

-Batch skeleton
PRMFL : \$UMCS/\$FILS.QC \$UMCS/\$FILS.YC QC, YC

-OLSD/DBD skeleton
PRMFL : \$UMCS/\$FILS.QG \$UMCS/\$FILS.YG QG, YG

-Reverse skeleton
PRMFL : \$UMCS/\$FILS.QR \$UMCS/\$FILS.YR QR, YR

-Client/Server skeleton
PRMFL : \$UMCS/\$FILS.QS \$UMCS/\$FILS.XS QS, XS

.GENERAL WORK FILES

File (FLR 166, CISZ = 2,048) W1

File (FLR 166, CISZ = 2,048) W2

File (FLR 166, CISZ = 2,048) W3

File (FLR 90, CISZ = 2,546) W4

File (FLR 308, CISZ = 2,816) W6

File (FLR 308, CISZ = 2,816) W7

File (FLR 308, CISZ = 2,816) W8

File (FLR 55, CISZ = 1,011) W9

File (FLR 100, CISZ = 6,144) WA

.SORT FILES

File Sn

	PAGE	70
STANDARD PROCEDURES		2
GPRT: GENERATION AND PRINTING		3
GPRT: DESCRIPTION OF STEPS		5

STRUCTURE OF THE PROCEDURE JCL

There are two ways of submitting the GPRT procedure to the GCOS8 Operating System:

- Using the 'JRN' command on TSS,
- Using the 'JOB' command on-line from VA Pac's GP screen.

When the GPRT procedure is submitted on-line, an additional step is initially performed in order to transmit the requests validated within the GP screen to the user request file.

The JCL corresponding to this step is defined in the following file:

&JCL/GPRB

Whatever the submission type (TSS or TP), the second step of the procedure processes the requests submitted by the user.

The JCL corresponding to this step is defined in the following file:

&JCL/GPRE

Within this step, the processing associated with a generation category (then with a monitor) can be recognized thanks to the structure described as follows:

- A label with the monitor code;
- The definition of a number of parameters (GLOBAL) necessary for the monitor to work properly;
- Call of a procedure that can detect if user requests need the monitor to be activated:

SELECT &JCL/DRVr

- Call of a procedure related to the given monitor:

SELECT &JCL/PACBx

STANDARD PROCEDURES	PAGE	71
GPRT: GENERATION AND PRINTING		2
GPRT: DESCRIPTION OF STEPS		3
		5

- Call of the PAF procedure if PAF operators are accepted by the generation category processed by the given monitor:

```
SELECT  &JCL/PAF
```

(This procedure is automatically executed when the related generator detects PAF operators in the generated entity.)

- Call of a procedure that detects the end of the given monitor's processing:

```
SELECT  &JCL/EMONI
```

This structure is repeated as many times as there are monitors in the GPRT procedure.

There are two monitors with a slightly different structure:

- PACBL processes the standard error message generation and have two more procedure calls right after the one related to the monitor. These calls are:

- . Call for the shift of error message file:

```
SELECT  &JCL/PACBLL
```

- . Call for the transformation of the revamped applications file to the TSS format:

```
SELECT  &JCL/PACBLH
```

- PACBK processes the Client/Server error message generation and have one more procedure call right after the one related to the monitor. This call is:

- . Call for rotation of the Client/Server error message file:

```
SELECT  &JCL/PACBKL
```

STANDARD PROCEDURES	PAGE	72
GPRT: GENERATION AND PRINTING		2
GPRT: DESCRIPTION OF STEPS		3
		5

EXECUTION PRINCIPLE OF THE PROCEDURE'S JOBS

When the GPRT procedure is submitted, the first monitor (PACBA) analyses user requests and creates a temporary file of DIS 'lud'. This file identifies all the monitors that have to process at least one request.

For each monitor, while there is still a request to be processed in the temporary file, the JCL/DRVR procedure analyses this file and either activates the monitor if it is concerned by the request or branches off to the end of the concerned processes on the &JCL/EMONI procedure.

When all the requests are processed, the last monitor is activated (PACBED), this monitor processes printing.

For example:

Program generation request (GCP)

STEP-1	PACBA	Monitor	Requests process
STEP-2	PACBE	Monitor	Screen requests analysis
STEP-3	PACBP	Monitor	Batch requests analysis
STEP-4	"		Program generation
STEP-5	PACBN	Monitor	User Manual requests analysis
STEP-6	PACBED	Monitor	Print process

CANCELLATION OF UNUSED MONITORS

The GPRT procedure contains a large number of SELECT clauses and parameters. Their processing takes some time during activity '00'.

In order to reduce the waiting time, it is advised to remove from the JCL the steps corresponding to unused generation categories.

To remove a category, delete the lines from the 'PACBx' label (monitor code associated to the category) up to the selection line of this monitor's 'EMONI' JCL.

MEANING OF THE PROCEDURE'S PARAMETERS

There are two types of parameters in this procedure:

. Procedure's general parameters:

- IMP Identifies the printer type and ensures the printing of the following reports:
 - = ASCII for an ASCII printer
 - = BCD for a BCD printer
- RMTA Identifies the remote workstation code for ASCII prints. The value of this parameter is set at the product's installation.
- RMTB Identifies the remote workstation code for BCD prints. The value of this parameter is set at the product's installation.
- JCL Identifies the entire code (UMC included) of the VA Pac JCLs catalog.

. Monitor's specific parameters:

- MONI Identifies the monitor's code. For each category of generation, the value of this parameter must be the code of the monitor that processes this category.
- LTPGx Defines the size of the memory that is necessary for a monitor to work (x = code of the generation category).
- LTCPx Defines the CPU time that is necessary for a monitor to work (x = code of the generation category).
- LTOUx Defines the SYSOUT size for a monitor's specific print (x = code of the generation category).
- TYPP Defines the generation type for the PAF processor.

STANDARD PROCEDURES	
GPRT: GENERATION AND PRINTING	
GPRT: DESCRIPTION OF STEPS	

2
3
5

MODIFICATION OF THE PROCEDURE'S PARAMETERS

All the parameters that users can modify are stored in the following file:
&JCL/PGPRT

They concern the workstations associated with each type of printer (ASCII, BCD), the CPU time allocated to each monitor and the sysout size associated with monitors that could process large printouts.

```
+-----+
! The value of these parameters will be kept after !
! the installation of a new VA Pac release.         !
+-----+
```

DEFINITION OF USER ENTRY POINTS

Some user entry points are authorized within the GPRE procedure (&JCL.GPRE).

The execution conditions of processing are automatically taken into account and depend on the insertion location within every monitor's procedures.

These execution conditions follow these rules:

- 1) Insertion of a processing right after the &JCL/DRVVR procedure.

This processing is automatically executed before a generation-print request associated with the selected monitor.

- 2) Insertion of a processing right before the &JCL/EMONI procedure.

This processing is automatically executed after the generation stage of a request associated with the selected monitor.

STANDARD PROCEDURES	
GPRT: GENERATION AND PRINTING	
GPRT: DESCRIPTION OF STEPS	

PAGE	75
	2
	3
	5

3) Particular case of standard error message generation.

User processing associated with the generation of error messages can be inserted before or after the file shift.
 In this case, insertion must respectively take place before or after the &JCL/PACBLL procedure of the PACBL monitor.

User processing associated with the generation of revamped applications must take place after the &JCL/PACBLH procedure of the PACBL monitor.

4) Particular case of Client/Server error message generation.

User processing associated with the generation of error messages can be inserted before or after the file shift.
 In this case, insertion must respectively take place before or after the &JCL/PACBKL procedure of the PACBK monitor.

RETRIEVAL OF USER PROCESSES

```

+-----+
! If you follow the insertion rules defined above, user !
! processes will be fully and automatically integrated in !
! future releases of your VA Pac system. !
+-----+

```

For this procedure to remain readable, it is highly recommended to insert user processes as procedure SELECTs.

STANDARD PROCEDURES	
GPRT: GENERATION AND PRINTING	
GPRT: DESCRIPTION OF STEPS	

LIST OF SWITCHES IN USE

Some switches are activated by the various procedures associated with a monitor.

NOTE: it is no longer necessary to test the value of these switches in order to condition the execution of the user processing. The insertion level only conditions their execution.

SWITCH-19 : No request.

SWITCH-20 : Programed ABORT for the GPRT procedure.

SWITCH-21 : Identifies a generation or print request for every monitor. For the generation requests, this switch remains to ON until the processing of the next monitor starts.

SWITCH-22 : All generation-print requests are processed and the &JCL/DRVR procedure branches off to the print processing monitor (PACBED) right away.

SWITCH-23 : For every concerned monitor, identifies the presence of PAF operators in the generated entity. Conditions the execution of the &JCL/PAF execution.

SWITCH-24 : For the standard error message processing monitor, identifies a request for the generation of revamped application. Conditions the execution of the &JCL/PACBLH procedure.

SWITCH-30 : In the set of processed requests, identifies at least one generation request.

SWITCH-35 : Manages, for the production environment, the TP8/GPRT concurrency for the (AB, XB) file.

STANDARD PROCEDURES	PAGE	77
GPRT: GENERATION AND PRINTING		2
GPRT: DESCRIPTION OF STEPS		3
		5

INTEGRATION OF THE PAF-PDM FUNCTION

Activating the PAF-PDM function modifies the execution of the GPRT chain for personalized documentation requests.

The PACBN step is reduced to a SELECT clause in the GDP74 or GDP85 JCL.

The GDP74 JCL corresponds to the execution of the H* module in real mode, and the GDP85 JCL corresponds to the execution of the run-unit in virtual mode.

In virtual mode, the run-unit is composed of a main module (PACBN) which dynamically calls sub-programs. The number of sub-programs called varies with the print request. A list of sub-programs is sent when PACBN is run via the DIRFC directive of the RUN command. This directive references the file with the FILE-CODE AA which corresponds to the ADRU JCL. At installation, this file contains the sub-programs which along with the principal module PACBN, constitute the most important part of the run-unit.

When creating a macro-command, this file forms a new PACBN sub-program with the help of the XPAF and XPDM procedures. The XPAF procedure updates the ADRU JCL and adds it to the list.

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 temporary 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.

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
GPRT: EXECUTION JCL

PAGE

78

2
3
6

2.3.6. GPRT: EXECUTION JCL

GPRT SUBMISSION VIA THE 'JOB' FUNCTION

The following JCL lines are to be entered on the Generation and Print Commands (GP) screen with 'C4' in the OPERATION CODE field.

```
$  USERID  $UMCB$PWB          )  
$  IDENT   $IDENT,$DEST.GPRT  )  LINE NUMBER < 600,000  
$  GLOBAL  USER=( $USER ),IMP=BCD )  
$  SELECT  $UMCU/$JCL.GPRB    )
```

```
-----  
$  SELECT  $UMCU/$JCL.GPRE     )  LINE NUMBER > 600,000  
$  ENDJOB
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$ IDENT $IDENT,$DEST.GPRT
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION AND PRINTING *
$ NOTE * * *
$ NOTE * SYMBOLICS *
$ NOTE * * *
$ NOTE * USER = USER CODE FOR FILES SUFFIX *
$ NOTE * * *
$ NOTE * FILE = NOTE IF BATCH SUBMISSION. *
$ NOTE * FILE IF TP SUBMISSION BY 'JOB'. *
$ NOTE * * *
$ NOTE * PRMFL = PRMFL IF BATCH SUBMISSION. *
$ NOTE * NOTE IF TP SUBMISSION BY 'JOB'. *
$ NOTE * * *
$ NOTE * ACCES = L IF REQUEST ' +AG' *
$ NOTE * Q OTHERWISE. *
$ NOTE * * *
$ NOTE * GDP = 74 IF MANUAL GENERATION IN REAL MODE *
$ NOTE * 85 IF MANUAL GENERATION IN VIRTUAL MODE *
$ NOTE * * *
$ NOTE * TP8 = 0 IF TP8 MONITOR *
$ NOTE * N IF DMIV-TP *
$ NOTE * * *
$ NOTE * IMP = ASCII IF ASCII PRINTING FORMAT *
$ NOTE * BCD IF BCD PRINTING FORMAT *
$ NOTE * * *
$ NOTE * JCL = STRING OF THE CATALOG CONTAINING *
$ NOTE * USER'S JCLS *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.GP&USER *
$ NOTE * * *
$ NOTE * INPUT SYNTAX *
$ NOTE * GENERATION AND PRINT COMMAND LINE *
$ NOTE * * *
$ NOTE *****
$ GLOBAL USER=( $USER ), FILE=NOTE , PRMFL=PRMFL , ACCES=Q , GDP=$GDP
$ GLOBAL JCL=( $UMCU/$JCL ) , TP8=$TP8 , IMP=BCD
$ SELECT &JCL/PGPRT
$ SELECT &JCL/GPRE
$ ENDJOB

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: USER PARAMETERIZATION       *
$      NOTE      *                                               *
$      NOTE      * SYMBOLICS                                       *
$      NOTE      *                                               *
$      NOTE      *   RMTA = STATION CODE FOR ASCII PRINT          *
$      NOTE      *                                               *
$      NOTE      *   RMTB = STATION CODE FOR BCD PRINT            *
$      NOTE      *                                               *
$      NOTE      *   LTCP* = CPU TIME ALLOCATED TO THE MONITOR    *
$      NOTE      *                                               *
$      NOTE      *   LTOU* = MONITOR SYSOUT SIZE                  *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$      GLOBAL    RMTA=( $RMTA ) ,RMTB=( $RMTB )
$      GLOBAL    LTCPA=( )
$      GLOBAL    LTCPB=20
$      GLOBAL    LTCPD=20
$      GLOBAL    LTCPE=20
$      GLOBAL    LTCPG=20
$      GLOBAL    LTCPK=20
$      GLOBAL    LTCPL=20 ,LTOUL=10K
$      GLOBAL    LTCPM=20 ,LTOUM=30K
$      GLOBAL    LTCPN=20 ,LTOUN=30K
$      GLOBAL    LTCPP=20
$      GLOBAL    LTCPQ=20
$      GLOBAL    LTCPR=20
$      GLOBAL    LTCPV=20
$      GLOBAL    LTCPCQ=20 ,LTOUCQ=30K
$      GLOBAL    LTCPED=20 ,LTOUED=50K

```


STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```
$      NOTE      *****
$      NOTE      * VisualAge Pacbase          *
$      NOTE      * =====                  *
$      NOTE      *                               *
$      NOTE      * FIRST PART OF THE GPRT PROCEDURE FOR 'JOB' *
$      NOTE      * FUNCTION SUBMISSION.        *
$      NOTE      *                               *
$      NOTE      * =====                  *
$      GLOBAL    JCL=( $UMCU/$JCL)
$      SELECT    &JCL.PGPRT
$ UTL8.  ***** GB CD ---> ASCII *****
$      UTL8
U8FD ME,TSS.
READ AA 1F WRITE ME.
$      FILE      ME,M1S,10L
$      DATA     AA
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                       *
$      NOTE      *                                             *
$      NOTE      * GENERATION-PRINT REQUESTS PROCESS             *
$      NOTE      * USED IN BATCH AND BY THE 'JOB' FUNCTION.      *
$      NOTE      *                                             *
$      NOTE      * *****
$ INPUT.        USER INPUT
$      SELECT    &JCL/INPUT
$ PACBA.        REQUESTS ANALYSIS
$      SELECT    &JCL/PACBA
$ PACBE.        DIALOG GENERATION
$      GLOBAL    MONI=PACBE
$      GLOBAL    TYPP=GE
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBE
$      SELECT    &JCL/PAF
$      SELECT    &JCL/EMONI
$ PACBP.        BATCH GENERATION
$      GLOBAL    MONI=PACBP
$      GLOBAL    TYPP=GP
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBP
$      SELECT    &JCL/PAF
$      SELECT    &JCL/EMONI
$ PACBN.        PERSONALIZED MANUAL GENERATION
$      GLOBAL    MONI=PACBN
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBN
$      SELECT    &JCL/EMONI
$ PACBL.        V7 ERROR MESSAGES GENERATION
$      GLOBAL    MONI=PACBL
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBL
$      SELECT    &JCL/PACBLL
$      SELECT    &JCL/PACBLH
$      SELECT    &JCL/EMONI
$ PACBG.        CLIENT GENERATION
$      GLOBAL    MONI=PACBG
$      GLOBAL    TYPP=GG
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBG
$      SELECT    &JCL/PAF
$      SELECT    &JCL/EMONI
$ PACBV.        SERVER GENERATION
$      GLOBAL    MONI=PACBV
$      GLOBAL    TYPP=GV
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBV
$      SELECT    &JCL/PAF
$      SELECT    &JCL/EMONI
$ PACBK.        C/S ERROR MESSAGES GENERATION
$      GLOBAL    MONI=PACBK
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBK
$      SELECT    &JCL/PACBKL
$      SELECT    &JCL/EMONI
$ PACBB.        DBD GENERATION
$      GLOBAL    MONI=PACBB
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBB
$      SELECT    &JCL/EMONI
$ PACBQ.        RELATIONAL DBD GENERATION
$      GLOBAL    MONI=PACBQ
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBQ
$      SELECT    &JCL/EMONI
$ PACBR.        REVERSE GENERATION
$      GLOBAL    MONI=PACBR
$      GLOBAL    TYPP=GR
$      SELECT    &JCL/DRVR
$      SELECT    &JCL/PACBR

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```
$      SELECT &JCL/PAF
$      SELECT &JCL/EMONI
$ PACBD.    DATA GENERATION
$      GLOBAL MONI=PACBD
$      SELECT &JCL/DRVR
$      SELECT &JCL/PACBD
$      SELECT &JCL/EMONI
$ PACBM.    U MANUALS AND V6 MES. GENERATION
$      GLOBAL MONI=PACBM
$      SELECT &JCL/DRVR
$      SELECT &JCL/PACBM
$      SELECT &JCL/EMONI
$ PACBED.   PRINTOUTS
$      SELECT &JCL/PACBED
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```
$      NOTE      *****
$      NOTE      * VisualAge Pacbase                      *
$      NOTE      * =====                                *
$      NOTE      *                                         *
$      NOTE      *   GENERATION-PRINT: MONITORS SEQUENCE DRIVER *
$      NOTE      *                                         *
$      NOTE      * *****                                *
$      DEFAULT   ESTEP=PACBED
$ DRIVER.
$      IF        22,&ESTEP
$      PROGRAM   RLHS,ON1,DUMP
$      LIMITS    ,30K
$      PRMFL     MO,R,S,$UMCS/$FILS.&MONI
$      FILE      DR,D1S
$      PRMFL     H*,R/C,R,$UMCS/$HSTAR.PACDRV
$      IF        20,ERROR
$      IF        /21,E&MONI
```

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
GPRT: EXECUTION JCL

PAGE

85

2
3
6

LABEL: END OF MONITOR PROCESSING

\$ E&(MONI).

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: PAF PREPROCESSOR *
$ NOTE * * *
$ NOTE *****
$ IF /23,EPAF&TYPP
$ GLOBAL PRMFLP=PRMFL,DATAP=NOTE
$ SELECT &JCL/PAFB
$ SELECT &JCL/PAFE
$ IF 20,ERROR
$ EPAF&(TYPP).
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: USER INPUT *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ DEFAULT USER=( $USER ), FILE=FILE, PRMFL=NOTE
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ &FILE MB,M1R
$ &PRMFL MB,R,S,$UMCU/$MB.GP&USER
$ FILE BM,C1S,1R
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: REQUESTS ANALYSIS           *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$ PACBA.
$      DEFAULT  USER=( $USER ) ,ACCES=Q
$      DEFAULT  LTCPA=( ) ,LTPGA=90K
$      DEFAULT  TP8=$TP8
$      SELECT   &JCL/CON&TP8
$      PROGRAM  RLHS,ONL,DUMP
$      LIMITS   &LTCPA,&LTPGA
$      PRMFL    1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    AG,&ACCES,R,$UMCB/$BASE.AG
$      PRMFL    XG,&ACCES,R,$UMCB/$BASE.XG
$      PRMFL    AB,&ECR,R,$UMCB/$BASE.AB
$      PRMFL    XB,&ECR,R,$UMCB/$BASE.XB
$      PRMFL    AC,Q,R,$UMCB/$BASE.AC
$      PRMFL    XC,Q,R,$UMCB/$BASE.XC
$      FILE     DR,D1SS,1R
$      FILE     QX,,100R
$      FILE     JG,J1SS,10R
$      FILE     MG,,10R
$      FILE     W1,,10R
$      FILE     KF,K4SS,10R
$      FILE     KU,K9SS,10R
$      FILE     YL,Y1SS,10R
$      FILE     YM,Y2SS,10R
$      FILE     ME,C1R
$      FILE     S1,,50R
$      SYSOUT   EI,ORG
$      SYSOUT   IH,ORG
$      PRMFL    H*,R/C,R,$UMCS/$HSTAR.PACBA
$      DATA    .U
FILE   FC/AN/,NBUF/16/,BFSZ/4096/
FILE   FC/BN/,NBUF/16/,BFSZ/4096/
FILE   FC/AR/,NBUF/16/,BFSZ/4096/
FILE   FC/BR/,NBUF/16/,BFSZ/4096/
$      IF       20,ERROR
$      IF       19,PACBED

```


STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: DIALOG GENERATION *
$ NOTE * * *
$ NOTE *****
$ PACBE .
$ DEFAULT USER=($USER)
$ DEFAULT LTCPE=20,LTPGE=210K
$ PROGRAM RLHS,ON1,DUMP
$ LIMITS &LTCPE,&LTPGE
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL QG,Q,R,$UMCS/$FILS.QG
$ PRMFL YG,Q,R,$UMCS/$FILS.YG
$ PRMFL AP,Q,R,$UMCB/$BASE.AP
$ PRMFL XP,Q,R,$UMCB/$BASE.XP
$ PRMFL GE,W,S,$UMCU/$FILG.GE&USER
$ FILE JG,J1SS
$ FILE KE,K3SS,10R
$ FILE EE,L3SS,10R
$ FILE W1,,20R
$ FILE W2,,20R
$ FILE YL,Y1SS
$ FILE YM,Y2SS
$ FILE S1,,50R
$ SYSOUT EI,ORG
$ PRMFL H*,R/C,R,$UMCS/$HSTAR.PACBE
$ DATA .U
FILE FC/AN/,NBUF/8/,BFSZ/4096/
FILE FC/BN/,NBUF/8/,BFSZ/4096/
FILE FC/AR/,NBUF/8/,BFSZ/4096/
FILE FC/BR/,NBUF/8/,BFSZ/4096/
$ IF 20,ERROR
$ IF /21,EPACBE

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: BATCH GENERATION             *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$ PACBP .
$      DEFAULT  USER=( $USER)
$      DEFAULT  LTCPP=20 ,LTPGP=140K
$      PROGRAM  RLHS,ON1 ,DUMP
$      LIMITS   &LTCPP,&LTPGP
$      PRMFL    1* ,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    QC,Q,R,$UMCS/$FILS.QC
$      PRMFL    YC,Q,R,$UMCS/$FILS.YC
$      PRMFL    AP,Q,R,$UMCB/$BASE.AP
$      PRMFL    XP,Q,R,$UMCB/$BASE.XP
$      PRMFL    GP,W,S,$UMCU/$FILG.GP&USER
$      FILE     JG,J1SS
$      PRMFL    GI,W,S,$UMCU/$FILG.GI&USER
$      FILE     KP,K7SS,10R
$      FILE     EP,L7SS,10R
$      FILE     W1,,20R
$      FILE     W2,,20R
$      FILE     W3,,20R
$      FILE     W4,,20R
$      FILE     YL,Y1SS
$      FILE     YM,Y2SS
$      FILE     S1,,50R
$      SYSOUT   EI,ORG
$      PRMFL    H* ,R/C,R,$UMCS/$HSTAR.PACBP
$      DATA    .U
FILE    FC/AN/ ,NBUF/16/ ,BFSZ/4096/
FILE    FC/BN/ ,NBUF/16/ ,BFSZ/4096/
FILE    FC/AR/ ,NBUF/16/ ,BFSZ/4096/
FILE    FC/BR/ ,NBUF/16/ ,BFSZ/4096/
$      IF      20,ERROR
$      IF      /21,EPACBP

```

STANDARD PROCEDURES

PAGE

91

GPRT: GENERATION AND PRINTING

2
3

GPRT: EXECUTION JCL

6

```
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: PERSONALIZED MANUAL GENERATION *
$ NOTE * * *
$ NOTE *****
$ PACBN.
$ DEFAULT GDP=$GDP
$ SELECT &JCL/GDP&GDP
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: MANUAL GENERATION IN REAL MODE *
$ NOTE * * *
$ NOTE *****
$ DEFAULT USER=( $USER ) , IMP=BCD , RMTA=( $RMTA ) , RMTB=( $RMTB )
$ DEFAULT LTCPN=20 , LTPGN=200K , LTOUN=30K
$ PROGRAM RLHS , ON1 , DUMP
$ LIMITS &LTCPN , &LTPGN
$ PRMFL 1 * , R / C , R , $UMCS / $SCHEMA . 1STAR
$ PRMFL AE , Q , R , $UMCB / $BASE . AE
$ PRMFL XE , Q , R , $UMCB / $BASE . XE
$ PRMFL AP , Q , R , $UMCB / $BASE . AP
$ PRMFL XP , Q , R , $UMCB / $BASE . XP
$ PRMFL AN , Q , R , $UMCB / $BASE . AN
$ PRMFL BN , Q , R , $UMCB / $BASE . BN
$ PRMFL AR , Q , R , $UMCB / $BASE . AR
$ PRMFL BR , Q , R , $UMCB / $BASE . BR
$ PRMFL GN , W , S , $UMCU / $FILG . GN & USER
$ PRMFL G6 , W , S , $UMCU / $FILG . G6 & USER
$ FILE JG , J1SS
$ FILE KN , K6SS , 10R
$ FILE EN , L6SS , 10R
$ FILE W1 , , 20R
$ FILE W2 , , 20R
$ FILE W6 , , 20R
$ FILE W7 , , 20R
$ FILE W8 , , 20R
$ FILE W9 , , 20R
$ FILE SV , , 20R
$ FILE YL , Y1SS
$ FILE YM , Y2SS
$ FILE S1 , , 50R
$ SYSOUT EI , ORG
$ FILE IN , M7S , 50L
$ PRMFL H * , R / C , R , $UMCS / $HSTAR . PACBN
$ DATA . U
FILE FC / AN / , NBUF / 8 / , BFSZ / 4096 /
FILE FC / BN / , NBUF / 8 / , BFSZ / 4096 /
FILE FC / AR / , NBUF / 8 / , BFSZ / 4096 /
FILE FC / BR / , NBUF / 8 / , BFSZ / 4096 /
$ IF 20 , ERROR
$ IF / 21 , EPACBN
$ GOTO N&IMP
$ NASCII .
$ ASCII-PRINT 132 CHAR .
$ CONVER
$ LIMITS , , , &LTOUN
$ FILE IN , M7R
$ SYSOUT OT , &RMTA
$ OUTPUT ASCII , MEDIA / 7
$ GOTO NEND
$ NBCD .
$ ASCII-PRINT 132 CHAR . --> BCD-PRINT 132 CHARACTERS **
$ CONVER
$ LIMITS , , , &LTOUN
$ FILE IN , M7R
$ SYSOUT OT , &RMTB
$ OUTPUT GBCD , MEDIA / 3
$ NEND .

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: MANUAL GENERATION IN VIRTUAL MODE *
$ NOTE * * *
$ NOTE *****
$ DEFAULT USER=( $USER ) , IMP=BCD , RMTA=( $RMTA ) , RMTB=( $RMTB )
$ DEFAULT LTCPN=20 , LTOUN=30K
$ RUN RUFIL=$UMCS/$RUNS.PACBN , DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR) ,
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ PRMFL AA , R , S , &JCL/ADRU
$ LIMITS &LTCPN
$ PRMFL AE , Q , R , $UMCB/$BASE.AE
$ PRMFL XE , Q , R , $UMCB/$BASE.XE
$ PRMFL AP , Q , R , $UMCB/$BASE.AP
$ PRMFL XP , Q , R , $UMCB/$BASE.XP
$ PRMFL AN , Q , R , $UMCB/$BASE.AN
$ PRMFL BN , Q , R , $UMCB/$BASE.BN
$ PRMFL AR , Q , R , $UMCB/$BASE.AR
$ PRMFL BR , Q , R , $UMCB/$BASE.BR
$ PRMFL GN , W , S , $UMCU/$FILG.GN&USER
$ PRMFL G6 , W , S , $UMCU/$FILG.G6&USER
$ PRMFL GS , Q , R , $UMCU/$FILU.GS
$ PRMFL YS , Q , R , $UMCU/$FILU.YS
$ FILE PA , , 100R
$ FILE YA , , 10R
$ FILE OG , 01SS , 10R
$ FILE OS , 02SS , 10R
$ FILE JG , J1SS
$ FILE KN , K6SS , 10R
$ FILE EN , L6SS , 10R
$ FILE W1 , , 20R
$ FILE W2 , , 20R
$ FILE W6 , , 20R
$ FILE W7 , , 20R
$ FILE W8 , , 20R
$ FILE W9 , , 20R
$ FILE SV , , 20R
$ FILE YL , Y1SS
$ FILE YM , Y2SS
$ FILE S1 , , 50R
$ SYSOUT DB , ORG
$ SYSOUT EI , ORG
$ FILE IN , M7S , 50L
$ DATA Uf
FC/PA/ NBUFF/10/
FC/YA/ NBUFF/10/
$ IF 20 , ERROR
$ IF /21 , EPACBN
$ GOTO N&IMP
$ NASCII .
$ ASCII-PRINT 132 CHAR .
$ CONVERT
$ LIMITS , , , &LTOUN
$ FILE IN , M7R
$ SYSOUT OT , &RMTA
$ OUTPUT ASCII , MEDIA/7
$ GOTO NEND
$ NBCD .
$ ASCII-PRINT 132 CHAR . --> BCD-PRINT 132 CHARACTERS **
$ CONVERT
$ LIMITS , , , &LTOUN
$ FILE IN , M7R
$ SYSOUT OT , &RMTB
$ OUTPUT GB CD , MEDIA/3
$ NEND .

```

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
GPRT: EXECUTION JCL

PAGE

94

2
3
6

'ADD RUN UNIT' DIRECTIVES IN VIRTUAL MODE

ADD_RU \$UMCS/\$RUNS.PACA90
ADD_RU \$UMCS/\$RUNS.PACABE
ADD_RU \$UMCS/\$RUNS.PACN25
ADD_RU \$UMCS/\$RUNS.PACN30
ADD_RU \$UMCS/\$RUNS.PACN33
ADD_RU \$UMCS/\$RUNS.PACN40
ADD_RU \$UMCS/\$RUNS.PACN50
ADD_RU \$UMCS/\$RUNS.PACN80
ADD_RU \$UMCS/\$RUNS.PACN90
ADD_RU \$UMCS/\$RUNS.PBBTST
ADD_RU \$UMCS/\$RUNS.SPABPB

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: V7 ERROR MESSAGES GENERATION *
$ NOTE * * *
$ NOTE *****
$ PACBL.
$ DEFAULT USER=( $USER ), IMP=BCD, RMTA=( $RMTA ), RMTB=( $RMTB )
$ DEFAULT LTCPL=20, LTPGL=160K, LTOUL=10K
$ PROGRAM RLHS, ON1, DUMP
$ LIMITS &LTCPL, &LTPGL
$ PRMFL 1*, R/C, R, $UMCS/$SCHEMA.1STAR
$ PRMFL AE, Q, R, $UMCB/$BASE.AE
$ PRMFL XE, Q, R, $UMCB/$BASE.XE
$ PRMFL AN, Q, R, $UMCB/$BASE.AN
$ PRMFL BN, Q, R, $UMCB/$BASE.BN
$ PRMFL AR, Q, R, $UMCB/$BASE.AR
$ PRMFL BR, Q, R, $UMCB/$BASE.BR
$ PRMFL AP, Q, R, $UMCB/$BASE.AP
$ PRMFL XP, Q, R, $UMCB/$BASE.XP
$ PRMFL LG, R, S, $UMCU/$FILG.LG&USER
$ PRMFL GL, W, S, $UMCU/$FILG.GL&USER
$ FILE GT, G1SS, 100R
$ FILE JG, J1SS
$ FILE LI, , 10R
$ FILE W1, , 20R
$ FILE W3, , 20R
$ FILE W4, , 20R
$ FILE YL, Y1SS
$ FILE YM, Y2SS
$ FILE S1, , 50R
$ SYSOUT EI, ORG
$ FILE IL, M5S, 50L
$ PRMFL H*, R/C, R, $UMCS/$HSTAR.PACBL
$ DATA .U
FILE FC/AN/, NBUF/16/, BFSZ/4096/
FILE FC/BN/, NBUF/16/, BFSZ/4096/
FILE FC/AR/, NBUF/16/, BFSZ/4096/
FILE FC/BR/, NBUF/16/, BFSZ/4096/
$ IF 20, ERROR
$ GOTO L&IMP
$ LASCII.
$ ASCII-PRINT 132 CHAR.
$ CONVER
$ LIMITS , , &LTOUL
$ FILE IN, M5R
$ SYSOUT OT, &RMTA
$ OUTPUT ASCII, MEDIA/7
$ GOTO LEND
$ LBCD.
$ ASCII-PRINT 132 CHAR. --> BCD-PRINT 132 CHARACTERS **
$ CONVER
$ LIMITS , , &LTOUL
$ FILE IN, M5R
$ SYSOUT OT, &RMTB
$ OUTPUT GBCD, MEDIA/3
$ LEND.
$ IF /21, EPACBL

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: SHIFT OF V7 ERROR MESSAGES *
$ NOTE * * *
$ NOTE *****
$ LPACBL.
$ SELECT &JCL/R&USER
```


STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: REVAMPING *
$ NOTE * * *
$ NOTE *****
$ HPACBL.
$ IF /24,EPACBL
$ UTL8
$ FILE A1,G1R
$ PRMFL A2,W,S,$UMCU/$FILG.GT&USER
$ U8FD A2,TSS.
$ READ A1 WRITE A2.
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                       *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: CLIENT GENERATION           *
$      NOTE      *                                               *
$      NOTE      * *****                                       *
$ PACBG.
$      DEFAULT  USER=( $USER)
$      DEFAULT  LTCPG=20,LTPGG=180K
$      PROGRAM  RLHS,ONL,DUMP
$      LIMITS   &LTCPG,&LTPGG
$      PRMFL    1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    QS,Q,R,$UMCS/$FILS.QS
$      PRMFL    XS,Q,R,$UMCS/$FILS.XS
$      PRMFL    AP,Q,R,$UMCB/$BASE.AP
$      PRMFL    XP,Q,R,$UMCB/$BASE.XP
$      PRMFL    GG,W,S,$UMCU/$FILG.GG&USER
$      FILE     JG,J1SS
$      FILE     KG,V1SS,10R
$      FILE     EG,W1SS,10R
$      FILE     W1,,20R
$      FILE     W2,,20R
$      FILE     YL,Y1SS
$      FILE     YM,Y2SS
$      FILE     S1,,50R
$      SYSOUT   EI,ORG
$      PRMFL    H*,R/C,R,$UMCS/$HSTAR.PACBG
$      DATA    .U
FILE    FC/AN/,NBUF/8/,BFSZ/4096/
FILE    FC/BN/,NBUF/8/,BFSZ/4096/
FILE    FC/AR/,NBUF/8/,BFSZ/4096/
FILE    FC/BR/,NBUF/8/,BFSZ/4096/
$      IF      20,ERROR
$      IF      /21,EPACBG

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: SERVER GENERATION           *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$ PACBV.
$      DEFAULT  USER=( $USER)
$      DEFAULT  LTCPV=20 ,LTPGV=220K
$      PROGRAM  RLHS,ONL ,DUMP
$      LIMITS   &LTCPV,&LTPGV
$      PRMFL    1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    QS,Q,R,$UMCS/$FILS.QS
$      PRMFL    XS,Q,R,$UMCS/$FILS.XS
$      PRMFL    AP,Q,R,$UMCB/$BASE.AP
$      PRMFL    XP,Q,R,$UMCB/$BASE.XP
$      PRMFL    GV,W,S,$UMCU/$FILG.GV&USER
$      FILE     JG,J1SS
$      FILE     KV,V2SS,10R
$      FILE     EV,W2SS,10R
$      FILE     W1,,20R
$      FILE     W2,,20R
$      FILE     YL,Y1SS
$      FILE     YM,Y2SS
$      FILE     S1,,50R
$      SYSOUT   EI,ORG
$      PRMFL    H*,R/C,R,$UMCS/$HSTAR.PACBV
$      DATA    .U
FILE    FC/AN/,NBUF/8/,BFSZ/4096/
FILE    FC/BN/,NBUF/8/,BFSZ/4096/
FILE    FC/AR/,NBUF/8/,BFSZ/4096/
FILE    FC/BR/,NBUF/8/,BFSZ/4096/
$      IF      20,ERROR
$      IF      /21,EPACBV

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

3

GPRT: EXECUTION JCL

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                       *
$      NOTE      *                                           *
$      NOTE      * GENERATION-PRINT: C/S ERROR MESSAGES GENERATION *
$      NOTE      *                                           *
$      NOTE      * *****                                       *
$ PACBK.
$      DEFAULT  USER=( $USER ) , IMP=BCD , RMTA=( $RMTA ) , RMTB=( $RMTB )
$      DEFAULT  LTCPK=20 , LTPGK=130K , LTOUK=10K
$      PROGRAM  RLHS , ON1 , DUMP
$      LIMITS   &LTCPK , &LTPGK
$      PRMFL    1* , R/C , R , $UMCS / $SCHEMA . 1STAR
$      PRMFL    AE , Q , R , $UMCB / $BASE . AE
$      PRMFL    XE , Q , R , $UMCB / $BASE . XE
$      PRMFL    AN , Q , R , $UMCB / $BASE . AN
$      PRMFL    BN , Q , R , $UMCB / $BASE . BN
$      PRMFL    AR , Q , R , $UMCB / $BASE . AR
$      PRMFL    BR , Q , R , $UMCB / $BASE . BR
$      PRMFL    AP , Q , R , $UMCB / $BASE . AP
$      PRMFL    XP , Q , R , $UMCB / $BASE . XP
$      PRMFL    LK , R , S , $UMCU / $FILG . LK&USER
$      PRMFL    GK , W , S , $UMCU / $FILG . GK&USER
$      FILE     JG , J1SS
$      FILE     LI , , 10R
$      FILE     WA , , 20R
$      FILE     YL , Y1SS
$      FILE     YM , Y2SS
$      FILE     S1 , , 50R
$      SYSOUT   EI , ORG
$      FILE     IK , M7S , 50L
$      PRMFL    H* , R/C , R , $UMCS / $HSTAR . PACBK
$      DATA    .U
FILE    FC/AN/ , NBUF/16/ , BFSZ/4096/
FILE    FC/BN/ , NBUF/16/ , BFSZ/4096/
FILE    FC/AR/ , NBUF/16/ , BFSZ/4096/
FILE    FC/BR/ , NBUF/16/ , BFSZ/4096/
$      IF      20 , ERROR
$      GOTO    K&IMP
$ KASCII.
$ ASCII-PRINT 132 CHAR.
$      CONVER
$      LIMITS  , , , &LTOUK
$      FILE    IN , M7R
$      SYSOUT  OT , &RMTA
$      OUTPUT  ASCII , MEDIA/7
$      GOTO    KEND
$ KBCD.
$ ASCII-PRINT 132 CHAR. --> BCD-PRINT 132 CHARACTERS **
$      CONVER
$      LIMITS  , , , &LTOUK
$      FILE    IN , M7R
$      SYSOUT  OT , &RMTB
$      OUTPUT  GBCD , MEDIA/3
$ KEND.
$      IF      /21 , EPACBK

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

3

GPRT: EXECUTION JCL

6

```
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: SHIFT OF C/S ERROR MESSAGES *
$ NOTE * * *
$ NOTE *****
$ LPACBK.
$ SELECT &JCL/S&USER
```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2
3

GPRT: EXECUTION JCL

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: DATABASE BLOCK GENERATION    *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$ PACBB.
$      DEFAULT  USER=($USER)
$      DEFAULT  LTCPB=20,LTPGB=140K
$      PROGRAM  RLHS,ONL,DUMP
$      LIMITS   &LTCPB,&LTPGB
$      PRMFL    1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    QG,Q,R,$UMCS/$FILS.QG
$      PRMFL    YG,Q,R,$UMCS/$FILS.YG
$      PRMFL    AP,Q,R,$UMCB/$BASE.AP
$      PRMFL    XP,Q,R,$UMCB/$BASE.XP
$      PRMFL    GB,W,S,$UMCU/$FILG.GB&USER
$      FILE     JG,J1SS
$      FILE     KB,K1SS,10R
$      FILE     W1,,20R
$      FILE     W2,,20R
$      FILE     YL,Y1SS
$      FILE     YM,Y2SS
$      FILE     S1,,50R
$      PRMFL    H*,R/C,R,$UMCS/$HSTAR.PACBB
$      SYSOUT   EI,ORG
$      DATA    .U
FILE    FC/AN/,NBUF/16/,BFSZ/4096/
FILE    FC/BN/,NBUF/16/,BFSZ/4096/
FILE    FC/AR/,NBUF/16/,BFSZ/4096/
FILE    FC/BR/,NBUF/16/,BFSZ/4096/
$      IF      20,ERROR
$      IF      /21,EPACBB

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: RELATIONAL DATABASE GENERATION *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$ PACBQ .
$      DEFAULT  USER=( $USER)
$      DEFAULT  LTCPQ=20, LTPGQ=160K
$      PROGRAM  RLHS, ON1, DUMP
$      LIMITS   &LTCPQ, &LTPGQ
$      PRMFL    1*, R/C, R, $UMCS/$SCHEMA.1STAR
$      PRMFL    AE, Q, R, $UMCB/$BASE.AE
$      PRMFL    XE, Q, R, $UMCB/$BASE.XE
$      PRMFL    AN, Q, R, $UMCB/$BASE.AN
$      PRMFL    BN, Q, R, $UMCB/$BASE.BN
$      PRMFL    AR, Q, R, $UMCB/$BASE.AR
$      PRMFL    BR, Q, R, $UMCB/$BASE.BR
$      PRMFL    AP, Q, R, $UMCB/$BASE.AP
$      PRMFL    XP, Q, R, $UMCB/$BASE.XP
$      PRMFL    GQ, W, S, $UMCU/$FILG.GQ&USER
$      FILE     EQ, E1SS, 10R
$      FILE     JG, J1SS
$      FILE     KQ, K0SS, 10R
$      FILE     YL, Y1SS
$      FILE     YM, Y2SS
$      PRMFL    H*, R/C, R, $UMCS/$HSTAR.PACBQ
$      SYSOUT   EI, ORG
$      DATA    .U
FILE   FC/AN/, NBUF/16/, BFSZ/4096/
FILE   FC/BN/, NBUF/16/, BFSZ/4096/
FILE   FC/AR/, NBUF/16/, BFSZ/4096/
FILE   FC/BR/, NBUF/16/, BFSZ/4096/
$      IF       20, ERROR
$      IF       /21, EPACBQ

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2
3

GPRT: EXECUTION JCL

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: REVERSE GENERATION          *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$ PACBR.
$      DEFAULT  USER=($USER)
$      DEFAULT  LTCPR=20,LTPGR=140K
$      PROGRAM  RLHS,ONL,DUMP
$      LIMITS   &LTCPR,&LTPGR
$      PRMFL    1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    QR,Q,R,$UMCS/$FILS.QR
$      PRMFL    YR,Q,R,$UMCS/$FILS.YR
$      PRMFL    AP,Q,R,$UMCB/$BASE.AP
$      PRMFL    XP,Q,R,$UMCB/$BASE.XP
$      PRMFL    GR,W,S,$UMCU/$FILG.GR&USER
$      FILE     KR,L1SS,10R
$      FILE     ER,E2SS,10R
$      FILE     JG,J1SS
$      FILE     W1,,20R
$      FILE     W2,,20R
$      FILE     YL,Y1SS
$      FILE     YM,Y2SS
$      FILE     S1,,50R
$      SYSOUT   EI,ORG
$      PRMFL    H*,R/C,R,$UMCS/$HSTAR.PACBR
$      DATA    .U
FILE    FC/AN/,NBUF/16/,BFSZ/4096/
FILE    FC/BN/,NBUF/16/,BFSZ/4096/
FILE    FC/AR/,NBUF/16/,BFSZ/4096/
FILE    FC/BR/,NBUF/16/,BFSZ/4096/
$      IF      20,ERROR

```


STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

GPRT: EXECUTION JCL

3

6

```

$      NOTE      *****
$      NOTE      * VisualAge Pacbase                               *
$      NOTE      * =====                                         *
$      NOTE      *                                               *
$      NOTE      * GENERATION-PRINT: DATA GENERATION             *
$      NOTE      *                                               *
$      NOTE      * *****                                         *
$ PACBD .
$      DEFAULT  USER=( $USER)
$      DEFAULT  LTCPD=20,LTPGD=140K
$      PROGRAM  RLHS,ONL,DUMP
$      LIMITS   &LTCPD,&LTPGD
$      PRMFL    1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    QG,Q,R,$UMCS/$FILS.QG
$      PRMFL    YG,Q,R,$UMCS/$FILS.YG
$      PRMFL    AP,Q,R,$UMCB/$BASE.AP
$      PRMFL    XP,Q,R,$UMCB/$BASE.XP
$      PRMFL    GD,W,S,$UMCU/$FILG.GD&USER
$      FILE     KD,K2SS,10R
$      FILE     KS,K8SS,10R
$      FILE     JG,J1SS
$      FILE     W1,,20R
$      FILE     W2,,20R
$      FILE     YL,Y1SS
$      FILE     YM,Y2SS
$      FILE     S1,,50R
$      SYSOUT   EI,ORG
$      PRMFL    H*,R/C,R,$UMCS/$HSTAR.PACBD
$      DATA    .U
FILE    FC/AN/,NBUF/16/,BFSZ/4096/
FILE    FC/BN/,NBUF/16/,BFSZ/4096/
FILE    FC/AR/,NBUF/16/,BFSZ/4096/
FILE    FC/BR/,NBUF/16/,BFSZ/4096/
$      IF      20,ERROR
$      IF      /21,EPACBD

```

STANDARD PROCEDURES

2

GPRT: GENERATION AND PRINTING

3

GPRT: EXECUTION JCL

6

```

$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: U MANUALS AND V6 LABELS GENERATION *
$ NOTE * * *
$ NOTE *****
$ PACBM.
$ DEFAULT USER=( $USER ), IMP=BCD, RMTA=( $RMTA ), RMTB=( $RMTB )
$ DEFAULT LTCPM=20, LTPGM=130K, LTOUM=30K
$ PROGRAM RLHS, ON1, DUMP
$ LIMITS &LTCPM, &LTPGM
$ PRMFL 1*, R/C, R, $UMCS/$SCHEMA.1STAR
$ PRMFL AE, Q, R, $UMCB/$BASE.AE
$ PRMFL XE, Q, R, $UMCB/$BASE.XE
$ PRMFL AN, Q, R, $UMCB/$BASE.AN
$ PRMFL BN, Q, R, $UMCB/$BASE.BN
$ PRMFL AR, Q, R, $UMCB/$BASE.AR
$ PRMFL BR, Q, R, $UMCB/$BASE.BR
$ PRMFL GM, W, R, $UMCU/$FILG.GM&USER
$ FILE JG, J1SS
$ FILE KM, K5SS, 10R
$ FILE W1, , 20R
$ FILE W2, , 20R
$ FILE YL, Y1SS
$ FILE YM, Y2SS
$ FILE S1, , 50R
$ SYSOUT EI, ORG
$ FILE IM, M6S, 50L
$ PRMFL H*, R/C, R, $UMCS/$HSTAR.PACBM
$ DATA .U
FILE FC/AN/, NBUF/16/, BFSZ/4096/
FILE FC/BN/, NBUF/16/, BFSZ/4096/
FILE FC/AR/, NBUF/16/, BFSZ/4096/
FILE FC/BR/, NBUF/16/, BFSZ/4096/
$ IF 20, ERROR
$ IF /21, EPACBM
$ GOTO M&IMP
$ MASCII.
$ ASCII-PRINT 136 CH. --> ASCII-PRINT 132 CHARACTERS **
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.UTI120
$ EXECUTE DUMP
$ FILE QC, M6R
$ FILE QD, N6S, 50L
$ MCONV.
$ CONVER
$ LIMITS , , , &LTOUM
$ FILE IN, N6R
$ SYSOUT OT, &RMTA
$ OUTPUT ASCII, MEDIA/7
$ GOTO MEND
$ MBCD.
$ ASCII-PRINT 136 CH. --> BCD-PRINT 136 CHARACTERS **
$ CONVER
$ LIMITS , , , &LTOUM
$ FILE IN, M6R
$ SYSOUT OT, &RMTB
$ OUTPUT GB CD, MEDIA/3
$ MEND.

```

STANDARD PROCEDURES

GPRT: GENERATION AND PRINTING

2

3

GPRT: EXECUTION JCL

6

```

$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * GENERATION-PRINT: GENERAL DOCUMENTATION PRINTING *
$ NOTE * * *
$ NOTE *****
$ PACBED.
$ DEFAULT IMP=BCD,RMTA=( $RMTA ),RMTB=( $RMTB )
$ DEFAULT LTCPED=20,LTPGED=120K,LTOUED=50K
$ PROGRAM RLHS,ON1,DUMP
$ LIMITS &LTCPED,&LTPGED
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE EQ,E1R
$ FILE KB,K1R
$ FILE KD,K2R
$ FILE KE,K3R
$ FILE KQ,K0R
$ FILE EE,L3R
$ FILE KF,K4R
$ FILE KM,K5R
$ FILE KN,K6R
$ FILE EN,L6R
$ FILE KP,K7R
$ FILE EP,L7R
$ FILE KS,K8R
$ FILE KU,K9R
$ FILE KR,L1R
$ FILE ER,E2R
$ FILE KG,V1R
$ FILE EG,W1R
$ FILE KV,V2R
$ FILE EV,W2R
$ FILE YL,Y1R
$ FILE YM,Y2R
$ FILE S1,,50R
$ SYSOUT EI,ORG
$ FILE ID,M9S,50L
$ SYSOUT IA,ORG
$ PRMFL H*,R/C,R,$UMCS/$HSTAR.PACBED
$ IF 20,ERROR
$ IF 19,END
$ GOTO Z&IMP
$ ZASCII.
$ ASCII-PRINT 136 CH. --> ASCII-PRINT 132 CHARACTERS **
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.UTI120
$ EXECUTE DUMP
$ FILE QC,M9R
$ FILE QD,N9S,50L
$ ZCONV.
$ CONVER
$ LIMITS , , ,&LTOUED
$ FILE IN,N9R
$ SYSOUT OT,&RMTA
$ OUTPUT ASCII,MEDIA/7
$ GOTO END
$ ZBCD.
$ ASCII-PRINT 136 CH. --> BCD-PRINT 136 CHARACTERS **
$ CONVER
$ LIMITS , , ,&LTOUED
$ FILE IN,M9R
$ SYSOUT OT,&RMTB
$ OUTPUT GB CD,MEDIA/3
$ END.
$ CONVER
$ DATA IN
**** GPRT - NORMAL END OF RUN ****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.

```

2.3.7. INTERFACE WITH GDT-PC

INTERFACE WITH GDT-PC

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

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

Example of control cards in front of program:

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

```
A      TITLE      TYPE : D      OPTION : P  
      CONTROL CARDS IN FRONT OF PROGRAM  
  
A NL DESCRIPTION OF THE CONTROL CARD      S PARM.R  
$$> -      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.

STANDARD PROCEDURES	PAGE	109
GPRT: GENERATION AND PRINTING		2
EMLD: LOADING OF USER-DEFINED ERROR MESSAGES		3
		8

2.3.8. EMLD: LOADING OF USER-DEFINED ERROR MESSAGES
2.3.8.1. 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.

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
EMLD: DESCRIPTION OF STEPS

PAGE

110

2
3
9

2.3.9. EMLD: DESCRIPTION OF STEPS

EMLD: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

INDEXED LOADING OF USER-DEFINED ERROR MESSAGES: PACL93

.Input files:
-Input transactions
 File MB
-Sequential user-defined error messages
 PRMFL : \$UMCU/\$FILG.LG&USER GL
-Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-VisualAge Pacbase error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Permanent output file:
-User-defined error messages, indexed
 PRMFL : \$UMCU/\$FILG.EM&USER EM
 \$UMCU/\$FILG.ME&USER ME
 (Blocking factor: 68)

.Output reports:
-Execution report
 SYSOUT IY
-Batch-procedure authorization option
 SYSOUT DD

.Return code:
-Switch-20
 1 - No batch-procedure authorization

STANDARD PROCEDURES
 GPRT: GENERATION AND PRINTING
 EMLD: EXECUTION JCL

PAGE

111

2
 3
 10

2.3.10. EMLD: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.EMLD
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * *
$ NOTE * LOADING OF USER DEFINED ERROR MESSAGES *
$ NOTE * *
$ NOTE * SYMBOLIC *
$ NOTE * *
$ NOTE * USER = USER CODE FOR FILE SUFFIX *
$ NOTE * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.EMLD *
$ NOTE * *
$ NOTE *****
$ GLOBAL USER=($USER)
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.EMLD
$ FILE BM,C1S,1R
$ PACL93.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PACL93
$ LIBRARY LA,LB
$ EXECUTE DUMP
$ LIMITS ,60K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL GL,R,S,$UMCU/$FILG.LG&USER
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL EM,L,R,$UMCU/$FILG.EM&USER
$ PRMFL ME,L,R,$UMCU/$FILG.ME&USER
$ FILE MB,C1R
$ SYSOUT IY,ORG
$ SYSOUT DD,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
**** EMLD - NORMAL END OF RUN ****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

	PAGE	112
STANDARD PROCEDURES		
GPRT: GENERATION AND PRINTING		2
EMUP: UPDATE OF USER-DEFINED ERROR MESSAGES		3
		11

2.3.11. EMUP: UPDATE OF USER-DEFINED ERROR MESSAGES
2.3.11.1. 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.

2.3.12. 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	!

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
EMUP: DESCRIPTION OF STEPS

PAGE

114

2
3
13

2.3.13. EMUP: DESCRIPTION OF STEPS

EMUP: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

UPDATE OF USER-DEFINED ERROR MESSAGES: PACL92

.Input files:

-Sequential user-defined error messages
PRMFL : \$UMCU/\$FILG.LG&USER GL
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-VisualAge Pacbase error messages
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Transaction file
File MB

.Permanent output file:

-User-defined error message indexed file
PRMFL : \$UMCU/\$FILG.EM&USER EM
\$UMCU/\$FILG.ME&USER ME
(Blocking factor: 68)

.Output reports:

-Transaction report
SYSOUT IU
-Error message report
SYSOUT IX
-Batch-procedure authorization option
SYSOUT DD

.Return code:

-Switch-20
1 - No batch-procedure authorization

STANDARD PROCEDURES
 GPRT: GENERATION AND PRINTING
 EMUP: EXECUTION JCL

2
 3
 14

2.3.14. EMUP: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.EMUP
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * USER'S ERROR MESSAGES UPDATING *
$ NOTE * * *
$ NOTE * SYMBOLIC *
$ NOTE * * *
$ NOTE * USER = USER CODE FOR FILE SUFFIX *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.EMUP *
$ NOTE * * *
$ NOTE *****
$ GLOBAL USER=($USER)
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.EMUP
$ FILE BM,C1S,1R
$ PACL92.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PACL92
$ EXECUTE DUMP
$ LIMITS ,65K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL GL,R,S,$UMCU/$FILG.LG&USER
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ FILE MB,C1R
$ PRMFL EM,L,R,$UMCU/$FILG.EM&USER
$ PRMFL ME,L,R,$UMCU/$FILG.ME&USER
$ SYSOUT IU,ORG
$ SYSOUT IX,ORG
$ SYSOUT DD,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
**** EMUP - NORMAL END OF RUN ****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

STANDARD PROCEDURES	PAGE	116
GPRT: GENERATION AND PRINTING		2
PPAF: PAF PRE-PROCESSOR		3
		15

2.3.15. PPAF: PAF PRE-PROCESSOR
2.3.15.1. 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.

2.3.16. 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.

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
PPAF: DESCRIPTION OF STEPS

PAGE 118
2
3
17

2.3.17. PPAF: DESCRIPTION OF STEPS

PPAF: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

PREPROCESSOR: PAFP10

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Input file:
-Generated programs
File EN

.Output files:
-Generated programs to be compiled
PRMFL : \$UMCU/\$FILG.&TYPP&USER SO

.Output report:
-Execution report
SYSOUT EP

STANDARD PROCEDURES	
GPRT: GENERATION AND PRINTING	
PPAF: DESCRIPTION OF STEPS	

PAGE	119
	2
	3
	17

USE OF THE JCL PARAMETERS

A program may be translated in the PAF preprocessor before being compiled, in one of three ways:

- Automatically within the GPRT procedure.
- After running the GPRT procedure.

This option only allows for the translation of one source program at a time. The source is submitted manually, via execution of the PPAF procedure.

```
The JCL parameters to be initialized are:&PRMFLP=PRMFL
&DATAP =NOTE
&USER  =USER (from the GPRT procedure)
&TYPP  =GP (for a BATCH program)
        =GE (for on-line program)
```

- By inserting a control card in front of/in back of the generated flow. The transformation of the program is only performed when the generated file is submitted to the loader.

```
The JCL parameters to be initialized are:&PRMFLP=NOTE
&DATAP =DATA
&USER  =USER (from the GPRT procedure)
&TYPP  =GP (for a BATCH program)
        =GE (for an on-line program)
```

In this case, the source code of the generated program must be enclosed by a '\$ ASCII' line (last 'before' line) and a '\$ ENX' line (first 'after' line).

Since the GPRT procedure includes a preprocessor call, it is necessary to eliminate the calls to the &JCL/PAF procedure within the following monitors: PACBE, PACBP, PACBG, PACBV, and PACBR, in order to implement one of the last two methods.

STANDARD PROCEDURES
GPRT: GENERATION AND PRINTING
PPAF: EXECUTION JCL

PAGE

120

2
3
18

2.3.18. PPAF: EXECUTION JCL

```
$ IDENT $IDENT,$DEST.PPAF
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * *
$ NOTE * PAF PREPROCESSOR *
$ NOTE * *
$ NOTE *****
$ GLOBAL PRMFLP=PRMFL,DATAP=NOTE,TYPP=GP,USER=( $USER)
$ SELECT $UMCU/$JCL.PAFB
$ SELECT $UMCU/$JCL.PAFE
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** PPAF - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB
```


	PAGE	121
STANDARD PROCEDURES		
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
PACX: INTRODUCTION		4
		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.

STANDARD PROCEDURES

PACX: EXTRACTION FROM THE VA PAC DATABASE

2

4

PACX: USER INPUT COMMON TO ALL EXTRACTORS

2

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 ! uuuuuuu ! 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)
-----

```

	PAGE	123
STANDARD PROCEDURES		2
PACX: EXTRACTION FROM THE VA PAC DATABASE		4
PACX: USER INPUT COMMON TO ALL EXTRACTORS		2

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.

STANDARD PROCEDURES	PAGE	124
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
EXLI: LIBRARY EXTRACTION		4
		3

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.

STANDARD PROCEDURES	PAGE	125
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
EXLI: USER INPUT		4
		4

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.

STANDARD PROCEDURES	PAGE	126
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
EXTR: ENTITY EXTRACTION		4
		5

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                       !
-----

```

STANDARD PROCEDURES

PACX: EXTRACTION FROM THE VA PAC DATABASE

EXTR: USER INPUT

2

4

6

First line cont'd

```

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

```

Second line (continuation line for selections and exceptions):

```

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

```

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

```

      _WIEX_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.

STANDARD PROCEDURES	PAGE	129
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
EXPJ: TRANSACTION EXTRACTION FROM THE JOURNAL		4
		7

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

Batch procedure access authorization option:

- . level 2 is required.

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

- . database access authorization level 4 is required.

STANDARD PROCEDURES

PACX: EXTRACTION FROM THE VA PAC DATABASE

EXPJ: USER INPUT

2

4

8

2.4.8. EXPJ: USER INPUT

EXPJ: USER INPUTUSER 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.

STANDARD PROCEDURES	PAGE	131
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
EXPU: EXTRACTION OF UNUSED ENTITIES FOR PURGE		4
		9

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.

	PAGE	132
STANDARD PROCEDURES		
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
EXPU: EXTRACTION OF UNUSED ENTITIES FOR PURGE		4
		9

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

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 defini-!  
! ! ! ! tion screens: !  
! ! ! 'U' ! In the indicated library only !  
! ! ! 'Z' ! In the indicated library and corres-!  
! ! ! ! ponding 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 'ENTITY 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.

	PAGE	135
STANDARD PROCEDURES		
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
EXPU: USER INPUT		4
		10

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.)

	PAGE	136
STANDARD PROCEDURES		
PACX: EXTRACTION FROM THE VA PAC DATABASE	2	
EXUE: EXTRACTION OF USER ENTITIES	4	
	11	

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

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 ! WLEX ! 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.

STANDARD PROCEDURES	
PACX: EXTRACTION FROM THE VA PAC DATABASE	
RMEN: RENAME/MOVE OF ENTITIES	

2
4
13

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 batch update procedure (UPDT or UPDP).

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 occu- !
! ! ! ! rrences of a UE or all Segments or !
! ! ! ! Reports of a Data Structure !
! ! ! ! (implicit option for 'RN' and 'RP') !
! 35 ! 3 ! ! For extraction of WorkStation enti- !
! ! ! ! ties: 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).

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

EXECUTION RULES

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

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

PRINTED OUTPUT

This procedure prints out the following:

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

RESULT

The output is a sequential file which contains update transactions:

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

NOTES:

The replacement of entities (RP) does not ensure data consistency. Thus, if you replace a Data Element with another one in a Segment, RMEN 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.

STANDARD PROCEDURES	PAGE	143
PACX: EXTRACTION FROM THE VA PAC DATABASE		2
RMEN: RECOMMENDATIONS AND RESTRICTIONS		4
		15

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.

	PAGE	144
STANDARD PROCEDURES		
PACX: EXTRACTION FROM THE VA PAC DATABASE	2	
RMEN: RECOMMENDATIONS AND RESTRICTIONS	4	
	15	

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.

	PAGE	145
STANDARD PROCEDURES		2
PACX: EXTRACTION FROM THE VA PAC DATABASE		4
RMEN: RECOMMENDATIONS AND RESTRICTIONS		15

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.

	PAGE	146
STANDARD PROCEDURES		
PACX: EXTRACTION FROM THE VA PAC DATABASE	2	
RMEN: RECOMMENDATIONS AND RESTRICTIONS	4	
	15	

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.

STANDARD PROCEDURES
PACX: EXTRACTION FROM THE VA PAC DATABASE
PACX: DESCRIPTION OF STEPS

PAGE

147

2
4
16

2.4.16. PACX: DESCRIPTION OF STEPS

PACX: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

EXTRACTION: PACX

This step extracts transactions according to user input.

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Erroneous transactions
PRMFL : \$UMCU/\$FILU.ARCH(0) PJ

.Input transaction file:
-User input
File MB

.Work files:
-User input
File (FLR 80, CISZ 512) BM
-EXPU work file
File (FLR 55, CISZ 6144) MM
-EXPJ work file
File (FLR 152, CISZ 2816) MJ
-RMEN work file
File (FLR 180, CISZ 4608) TE
-RMEN work file
File (FLR 12, CISZ 4088) RE
-RMEN work file
File (FLR 167, CISZ 9413) RM

-Extracted transactions
File (FLR 167, CISZ 9413) WD
-Multi-layered Extractor work file
File (FLR 112, CISZ 3072, UIND) XX, XY

STANDARD PROCEDURES

PACX: EXTRACTION FROM THE VA PAC DATABASE

2

PACX: DESCRIPTION OF STEPS

4

16

.Output files:

-Extracted transactions for UPDT
 PRMFL : \$UMCU/\$MV.UPDT MV

-Extracted transactions for REOR (EXPU)
 PRMFL : \$UMCU/\$MV.REOR MR

-Extracted transactions for UPDP
 PRMFL : \$UMCU/\$MV.UPDP GY

-Extracted transactions for CPSN
 PRMFL : \$UMCU/\$FILU.MACPSN (ou SLCPSN) TD

-Extracted transactions for EXUE
 PRMFL : \$UMCU/\$MV.EXUE UE

.Output reports:

-General printout of the program stream
 SYSOUT IA

-List of errors on input transactions
 SYSOUT DD

-Summary reports on extractions

SYSOUT EE

SYSOUT EP

SYSOUT EQ

SYSOUT EZ

.Sort file(s):

File S1, S2, S3, S4

.Return codes:

0 - No error

4 - Error in user input (specified in EE)
 or
 EXTR/EXUE - problem during extraction
 (specified in EZ)

8 - Error on '*'-line (specified in DD)
 or
 EXLI - Database unavailable

STANDARD PROCEDURES

PACX: EXTRACTION FROM THE VA PAC DATABASE

PACX: EXECUTION JCL

2

4

17

2.4.17. PACX: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.PACX
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * EXTRACTIONS *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.PACX *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.PJO
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.PACX
$ FILE BM,C1S,1R
$ PACX.
$ RUN RUFIL=$UMCS/$RUNS.PACX,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ DATA AA
ADD_RU $UMCS/$RUNS.PACA90
ADD_RU $UMCS/$RUNS.PACABE
ADD_RU $UMCS/$RUNS.PACCTL
ADD_RU $UMCS/$RUNS.PACFGY
ADD_RU $UMCS/$RUNS.PACFMB
ADD_RU $UMCS/$RUNS.PACFTD
ADD_RU $UMCS/$RUNS.PACHOI
ADD_RU $UMCS/$RUNS.PACSJO
ADD_RU $UMCS/$RUNS.PACSMD
ADD_RU $UMCS/$RUNS.PACSPU
ADD_RU $UMCS/$RUNS.PACSRM
ADD_RU $UMCS/$RUNS.PACS30
ADD_RU $UMCS/$RUNS.PACS40
ADD_RU $UMCS/$RUNS.PACS50
ADD_RU $UMCS/$RUNS.PACS60
ADD_RU $UMCS/$RUNS.PACS75
ADD_RU $UMCS/$RUNS.PACS80
ADD_RU $UMCS/$RUNS.SPABPB
$ DATA Uf
FC/XX/ NBUFF/10/
FC/XY/ NBUFF/10/
$ LIMITS 20,,,100K
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL PJ,Q,R,&PJI
$ PRMFL GY,W,R,$UMCU/$MV.UPDP
$ PRMFL MR,W,S,$UMCU/$MV.REOR
$ PRMFL MV,W,S,$UMCU/$MV.UPDT
$ PRMFL TD,W,R,$UMCU/$FILU.MACPSN
$ C. PRMFL TD,W,R,$UMCU/$FILU.SLCPSN
$ PRMFL UE,W,S,$UMCU/$MV.EXUE
$ FILE MB,C1R
$ FILE BM,,10R
$ FILE MJ,,100R
$ FILE MM,,10R
$ FILE RE,,10R
$ FILE RM,,10R
$ FILE TE,,10R
$ FILE WD,,10R
$ FILE XX,,100R
$ FILE XY,,10R

```

STANDARD PROCEDURES

PACX: EXTRACTION FROM THE VA PAC DATABASE

PACX: EXECUTION JCL

2

4

17

```
$ FILE S1,,100R
$ FILE S2,,100R
$ FILE S3,,100R
$ FILE S4,,100R
$ SYSOUT DD,ORG
$ SYSOUT ED,ORG
$ SYSOUT EE,ORG
$ SYSOUT EI,ORG
$ SYSOUT EP,ORG
$ SYSOUT EQ,ORG
$ SYSOUT EZ,ORG
$ SYSOUT IA,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** NORMAL END OF RUN = PACX *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB
```

VisualAge Pacbase - Operations Manual	PAGE	151
BATCH PROCEDURES: USER'S GUIDE		
PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION		3

3. PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION	PAGE	152
XPAF: EXTRACTION MASTER PATH		3
XPAF: INTRODUCTION		1

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 .PPTX 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.

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION	PAGE	153
XPAF: EXTRACTION MASTER PATH		3
XPAF: INTRODUCTION		1
		1

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) !
!-----!
! 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 occurrence's !
! ! ! ! ! use in the sub-network. !
! ! ! ! ! No update of GS if presence or use. !
!-----!

```

EXAMPLES

```

*user passwordBIB
EX7EEXT001 UPDATE
*user passwordBIB
EX7EEXT002

```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
XPAF: EXTRACTION MASTER PATH
XPAF: DESCRIPTION OF STEPS

PAGE

155

3
1
3

3.1.3. XPAF: DESCRIPTION OF STEPS

XPAF: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

ACCESS AND VALIDATION: PTEX30

.Input files:
-VA Pac error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR

.Input transaction file:
-User input
File MB

.Permanent input file:
-Variable skeleton-file
PRMFL : \$UMCS/\$FILS.QP \$UMCS/\$FILS.YP QP, YP

.Permanent input/output file:
-Extraction Paths
PRMFL : \$UMCU/\$FILU.GS \$UMCU/\$FILU.YS GS, YS

.Output file:
-Summary passed on to printing program
File ED
-Temporary generated source
File GP

.Output report:
-Execution report
SYSOUT DD

.Sort file(s):
File S1

GENERATION OF THE COMPILATION JCL: PTEX31

.Input files:
-JCL for macro-command creation
PRMFL : \$UMCU/\$JCL.SPWM IM
-JCL for user extractor creation
PRMFL : \$UMCU/\$JCL.SPWX&GDP IX
-Information on generated entity
File IE

.Output files:
-Beginning of JCL to be spawned
File OB
-End if JCL to be spawned
File OE

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
 XPAF: EXTRACTION MASTER PATH
 XPAF: DESCRIPTION OF STEPS

3
 1
 3

EXTRACTION GENERATION: PTEX80

.Permenant input file:
 -Fixed skeleton file
 PRMFL : \$UMCS/\$FILS.SF SF

.Input file:
 -Source file generated by PTEX30
 File GP

.Output file:
 -Generated source to be translated
 File ST

FORMATTING THE JCL TO BE SPAWNED: UTL8

.Input file:
 -JCLs to be merged
 File IN

.Output file:
 -JCL to be spawned
 File OU

SPAWNING THE COMPILATION JCL: PTUJOB

.Input files:
 -JCL to be spawned
 File JI
 -SPAWN options
 PRMFL : \$UMCU/\$MB.SP&USER OP

PTEX PRINTING: PTEXD0

.Input files:
 -VA Pac error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
 -PTEX30 report
 File ED

.Permanent input/output file:
 PRMFL : \$UMCU/\$FILU.GS \$UMCU/\$FILU.YS GS, YS

.Output report:
 -Validation report
 SYSOUT RD

.Sort file(s):
 File S1

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
 XPAF: EXTRACTION MASTER PATH
 XPAF: EXECUTION JCL

3
 1
 4

3.1.4. XPAF: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.XPAF
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * VALIDATING AN EXTRACTION MASTER PATH *
$ NOTE * * *
$ NOTE * SYMBOLICS *
$ NOTE * * *
$ NOTE * USER = USER CODE FOR SPAWNING OPTIONS *
$ NOTE * FILE SUFFIX *
$ NOTE * * *
$ NOTE * GDP = 74 IF MANUAL GENERATION IN REAL MODE *
$ NOTE * 85 IF MANUAL GENERATION IN VIRTUAL MODE *
$ NOTE * * *
$ NOTE * IMP = ASCII IF ASCII PRINT FORMAT *
$ NOTE * BCD IF BCD PRINT FORMAT *
$ NOTE * * *
$ NOTE * RMTA = STATION CODE FOR ASCII PRINT *
$ NOTE * * *
$ NOTE * RMTB = STATION CODE FOR BCD PRINT *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.XPAF *
$ NOTE * * *
$ NOTE *****
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ GLOBAL USER=($USER),GDP=$GDP
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.XPAF
$ FILE BM,C1S,1R
$ PTEX30.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTEX30
$ EXECUTE DUMP
$ LIMITS ,160K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL QP,Q,R,$UMCS/$FILS.QP
$ PRMFL YP,Q,R,$UMCS/$FILS.YP
$ PRMFL GS,W,R,$UMCU/$FILU.GS
$ PRMFL YS,W,R,$UMCU/$FILU.YS
$ FILE ED,D1S,10R
$ FILE GP,G1S,10R
$ FILE MB,C1R
$ FILE ZE,I1S,10R
$ FILE S1,,100R
$ SYSOUT EI,ORG
$ SYSOUT DD,ORG
$ IF 20,ERROR
$ IF 30,PTEXD0
$ PTEX31.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTEX31
$ EXECUTE DUMP

```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION

XPAF: EXTRACTION MASTER PATH

3

1

XPAF: EXECUTION JCL

4

```

$      LIMITS      ,30K
$      PRMFL      IM,R,S,$UMCU/$JCL.SPWM
$      PRMFL      IX,R,S,$UMCU/$JCL.SPWX&GDP
$      FILE       IE,I1R
$      FILE       OB,O1S,10L
$      FILE       OE,O2S,10L
$ PTEX80.
$      OPTION     CBL74
$      LIBRARY    LA
$      SELECT     $UMCS/$OBJBT.PTEX80
$      EXECUTE    DUMP
$      LIMITS    ,30K
$      PRMFL     SF,R,R,$UMCS/$FILS.SF
$      FILE      GP,G1R
$      FILE      ST,P1S,50L
$      SYSOUT    EI,ORG
$      IF        20,PTEXD0
$ UTL8.
$      UTL8
$      FILE      IN,O1
$      FILE      ",P1
$      FILE      ",O2
$      FILE      OU,O3S,100L
$      READ      IN 1F WRITE OU.
$ SPAWN.
$      SELECT     $UMCU/$JCL.SPWN
$ PTEXD0.
$      OPTION     CBL74
$      LIBRARY    LA,LB
$      SELECT     $UMCS/$OBJBT.PTEXD0
$      EXECUTE    DUMP
$      LIMITS    ,80K
$      PRMFL     1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL     LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$      PRMFL     LA,R/C,R,$UMCS/$FILS.OBJLIB
$      PRMFL     AE,Q,R,$UMCB/$BASE.AE
$      PRMFL     XE,Q,R,$UMCB/$BASE.XE
$      PRMFL     GS,W,R,$UMCU/$FILU.GS
$      PRMFL     YS,W,R,$UMCU/$FILU.YS
$      FILE      ED,D1R
$      FILE      S1,,100R
$      SYSOUT    EI,ORG
$      FILE      RD,E1S,10L
$      GOTO      P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$      CONVER
$      LIMITS    ,,10K
$      FILE      IN,E1R
$      SYSOUT    OT,&RMTB
$      OUTPUT    GBCD,MEDIA/3
$      GOTO      END
$ PASCII.
$ ASCII-PRINT 132 CH.
$      CONVER
$      LIMITS    ,,10K
$      FILE      IN,E1R
$      SYSOUT    OT,&RMTA
$      OUTPUT    ASCII,MEDIA/7
$ END.
$      CONVER
$      DATA     IN
***** XPAF - NORMAL END OF RUN *****
$      SYSOUT    OT,ORG
$      OUTPUT    MEDIA/03
$ ERROR.
$      ENDJOB

```

3.1.5. XPAF: CREATION OF A MACRO-COMMAND

XPAF: CREATION OF A MACRO-COMMAND (DESCRIPTION OF STEPS)

CREATION OF THE RUN-UNIT: FILSYS

This step creates the envelope of the run-unit.

PRMFL: \$UMCU/\$FILX/\$RUNM.<<CGI>>

PRE-PROCESSING: PAFP10

.Permanent input files:

-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Input file:

-Source code of the macro-command
DATA (from the XPAF procedure) EN

.Output file:

-Program to be compiled
File SO

.Output report

-Execution report
SYSOUT EP

COMPILATION: CBL85

.Input file:

-Program to be compiled
File S*

.Output file:

-Object module
File O*

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
XPAF: EXTRACTION MASTER PATH
XPAF: CREATION OF A MACRO-COMMAND

PAGE

160

3
1
5

LINK-EDIT: LKED

.Input file:
-Object module
File O*

.Output file:
-Run-unit of the macro-command
PRMFL : \$UMCU/\$FILX/\$RUNM.<<CGI>> V*

COPY OF THE RUN-EDIT INSTRUCTIONS: UTL8

.Input file:
-Run-unit instructions file
PRMFL: \$UMCU/\$JCL.ADRU IN

.Output file:
-Copy
File OU

INSTRUCTIONS JCL UPDATING: PTUADR

.Input files:
-Run-unit instructions file
File IA
-Macro-command cobol source
File IC
-Macro-command creation JCL
PRMFL: \$UMCU/\$JCL.SPWM IM

.OUTPUT FILE
-Run-unit instructions file
PRMFL: \$UMCU/\$JCL.ADRU OA

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION

XPAF: EXTRACTION MASTER PATH

3

1

XPAF: CREATION OF A MACRO-COMMAND

5

```

$ IDENT $IDENT,$DEST.RUNM
$ USERID $UMCU$PWU
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * COMPILATION OF A MACRO-COMMAND *
$ NOTE * AND CREATION OF THE ASSOCIATED RUN-UNIT *
$ NOTE * INSTRUCTIONS JCL UPDATING (ADD_RU) *
$ NOTE * * *
$ NOTE * THIS PROCEDURE IS SPAWNED BY THE *
$ NOTE * XPAF PROCEDURE *
$ NOTE * * *
$ NOTE *****
$ FILSYS
USERID $UMCU$PWU
IGNORE ERRS
FC $UMCU/$FILX/$RUNM.<<CGI>>,
    LLINKS/1000,2000/,MODE/RAND/
$ PAFP10.
$ DEFAULT LUD=P1
$ OPTION CBL74
$ LIBRARY LA, LB
$ SELECT $UMCS/$OBJBT.PAFP10
$ EXECUTE DUMP
$ LIMITS 20,200K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ FILE SO,P2S,50L
$ SYSOUT EP,ORG
$ DATA EN
$ ASCII
$ ENX
$ IF 20,ERROR
$ COMPILE.
$ CBL85 COBOL74,MAIN,MSDATA, SORT_MEM=512
$ LIMITS 10,,,50K
$ FILE S*,P2S
$ FILE O*,O1S,50R
$ LINK.
$ LKED FORM
R -N_M -N_M_L -N_S_L
L -L CBL85
GRU -N <<CGI>>
CH -DATA 2048K -DESC 1K
I_O -FC O*
V -E <<CGI>>_ENTDEF
$ PRMFL V*,W,R,$UMCU/$FILX/$RUNM.<<CGI>>
$ FILE O*,O1R
$ UTL8.
$ UTL8
$ PRMFL IN,R,S,$UMCU/$JCL.ADRU
$ FILE OU,I1S,10L
$ READ IN 1F WRITE OU.
$ PTUADR.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTUADR
$ EXECUTE DUMP
$ LIMITS ,30K
$ FILE IA,I1
$ FILE IC,P2
$ PRMFL IM,R,S,$UMCU/$JCL.SPWM
$ PRMFL OA,W,S,$UMCU/$JCL.ADRU
$ END.
$ CONVER
$ DATA IN

```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
XPAF: EXTRACTION MASTER PATH
XPAF: CREATION OF A MACRO-COMMAND

PAGE

162

3
1
5

```
***** RUNM - NORMAL END OF RUN *****  
$      SYSOUT  OT,ORG  
$      OUTPUT  MEDIA/03  
$ ERROR.  
$      ENDJOB
```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION	PAGE	163
XPAF: EXTRACTION MASTER PATH		3
XPAF: CREATION OF A GENERAL-PURPOSE EXTRACTOR		1
		6

3.1.6. XPAF: CREATION OF A GENERAL-PURPOSE EXTRACTOR

CREATION OF A GENERAL-PURPOSE EXTRACTOR PRINCIPLE

This procedure is spawned by the XPAF procedure and enables the creation, according to the option chosen, of a generalized extractor in the form of either an object (COBOL-74) or a run-unit (COBOL-85).

The JCL's parameter settings are removed by the PTEX31 program of the XPAF procedure. The JCL is executed by the PTUJOB program.

CREATION OF AN EXTRACTOR (COBOL 74) - DESCRIPTION OF STEPS

CREATION OF THE OBJECT MODULE: FILSYS

This step creates the envelope of the object.

PRMFL: \$UMCU/\$FILX/\$074.<<CGI>>

PRE-PROCESSING: PAFP10

```
.Permanent input files:
-Data file
  PRMFL : $UMCB/$BASE.AR $UMCB/$BASE.BR      AR, BR
-Index file
  PRMFL : $UMCB/$BASE.AN $UMCB/$BASE.BN      AN, BN
-Error message file
  PRMFL : $UMCB/$BASE.AE $UMCB/$BASE.XE      AE, XE

.Input file:
-Source code of the generalized
  extractor from the XPAF procedure
  DATA                                     EN

.Output file:
-Program to be compiled
  File                                     SO

.Output report:
-Execution report
  SYSOUT                                    EP
```

COMPILATION: CBL74

```
.Input file:
-Program to be compiled
  File                                     S*

.Output file:
-Object module
  PRMFL : $UMCU/$FILX/$074.<<CGI>>        C*
```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION

XPAF: EXTRACTION MASTER PATH

3

1

XPAF: CREATION OF A GENERAL-PURPOSE EXTRACTOR

6

```

$ IDENT $IDENT,$DEST.OBJX
$ USERID $UMCU$PWU
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * COMPILATION OF A USER EXTRACTION PROGRAM *
$ NOTE * AND CREATION OF THE ASSOCIATED OBJECT *
$ NOTE * * *
$ NOTE * THIS PROCEDURE IS SPAWNED BY THE *
$ NOTE * XPAF PROCEDURE *
$ NOTE * * *
$ NOTE *****
$ FILSYS
USERID $UMCU$PWU
IGNORE ERRS
FC $UMCU/$FILX/$074.<<CGI>>,
  LLINKS/1000,2000/,MODE/SEQ/
$ PAFP10.
$ DEFAULT LUD=P1
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PAFP10
$ EXECUTE DUMP
$ LIMITS 20,200K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ FILE SO,P2S,50L
$ SYSOUT EP,ORG
$ DATA EN
$ ASCII
$ ENX
$ IF 20,ERROR
$ COMPILE.
$ CBL74 DECK,COPY,XREF,MAP,PMAP
$ LIMITS 10,250K,,50K
$ FILE S*,P2R
$ PRMFL C*,W,S,$UMCU/$FILX/$074.<<CGI>>
$ END.
$ CONVER
$ DATA IN
***** OBJX - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

	PAGE	166
PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION		3
XPAF: EXTRACTION MASTER PATH		1
XPAF: CREATION OF A GENERAL-PURPOSE EXTRACTOR		6

CREATION OF AN EXTRACTOR (COBOL-85) - DESCRIPTION OF STEPS

CREATION OF THE OBJECT MODULE: FILSYS

This step creates the envelope of the run-unit.

PRMFL: \$UMCU/\$FILX/\$RUNX.<<CGI>>

PRE-PROCESSING: PAFP10

.Permanent input files:

	-Data file	
PRMFL :	\$UMCB/\$BASE.AR \$UMCB/\$BASE.BR	AR, BR
	-Index file	
PRMFL :	\$UMCB/\$BASE.AN \$UMCB/\$BASE.BN	AN, BN
	-Error message file	
PRMFL :	\$UMCB/\$BASE.AE \$UMCB/\$BASE.XE	AE, XE

.Input file:

	-Source code of the generalized extractor from the XPAF procedure	
DATA		EN

.Output file:

	-Program to be compiled	
File		SO

.Output report:

	-Execution report	
SYSOUT		EP

COMPILATION: CBL85

.Input file:

	-Program to be compiled	
File		S*

.Output file:

	-Object module	
File		O*

LINK-EDIT: LKED

.Input file:

	-Object module	
File		O*

.Output file:

	-Run-unit of the extraction	
PRMFL :	\$UMCU/\$FILX/\$RUNX.<<CGI>>	V*

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION

XPAF: EXTRACTION MASTER PATH

XPAF: CREATION OF A GENERAL-PURPOSE EXTRACTOR

3

1

6

```

$ IDENT $IDENT,$DEST.RUNX
$ USERID $UMCU$PWU
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * COMPILATION OF A USER EXTRACTION PROGRAM *
$ NOTE * AND CREATION OF THE ASSOCIATED RUN-UNIT *
$ NOTE * * *
$ NOTE * THIS PROCEDURE IS SPAWNED BY THE *
$ NOTE * XPAF PROCEDURE *
$ NOTE * * *
$ NOTE *****
$ FILSYS
USERID $UMCU$PWU
IGNORE ERRS
FC $UMCU/$FILX/$RUNX.<<CGI>>,
  LLINKS/1000,2000/,MODE/RAND/
$ PAFP10.
$ DEFAULT LUD=P1
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PAFP10
$ EXECUTE DUMP
$ LIMITS 20,200K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ FILE SO,P2S,50L
$ SYSOUT EP,ORG
$ DATA EN
$ ASCII
$ ENX
$ IF 20,ERROR
$ COMPILE.
$ CBL85 COBOL74,MAIN,SORT_MEM=512
$ LIMITS 10,,,50K
$ FILE S*,P2R
$ FILE O*,O1S,50R
$ LINK.
$ LKED FORM
R -N_M -N_M_L -N_S_L
L -L CBL85
GRU -N <<CGI>>
CH -DATA 2048K -DESC 1K
I_O -FC O*
$ PRMFL V*,W,R,$UMCU/$FILX/$RUNX.<<CGI>>
$ FILE O*,O1R
$ END.
$ CONVER
$ DATA IN
***** RUNX - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION	PAGE	168
XPDM: MASTER OUTLINE		3
XPDM: INTRODUCTION		2
		1

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 ! !uuuuuuu! 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

TRANSACTION RECOGNITION: PTU001

EXTRACTION OF MASTER OUTLINE: PTED30

.Input files:
-Error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR

.Input transaction file:
-User input
File MB

.Permanent input/output file:
-Extraction paths
PRMFL : \$UMCU/\$FILU.GS \$UMCU/\$FILU.YS GS, YS

.Output files:
-Report passed on to printing program
File ED
-GS-update preparation
File TG

.Output report:
-Execution report
SYSOUT DD

GS UPDATE AND PRINTING OF THE MASTER OUTLINE: PTED60

.Input files:
-VA Pac error messages
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Print file
File ED
-GS-update preparation
File TG

.Permanent output file:
-Extraction Paths
PRMFL : \$UMCU/\$FILU.GS \$UMCU/\$FILU.YS GS, YS

.Output report:
-Execution report
SYSOUT GP

.Sort file(s):
File S1

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
 XPDM: MASTER OUTLINE
 XPDM: EXECUTION JCL

3
 2
 4

3.2.4. XPDM: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.XPDM
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * VALIDATING THE MASTER OUTLINE *
$ NOTE * * *
$ NOTE * SYMBOLICS *
$ NOTE * * *
$ NOTE * IMP = ASCII IF PRINTING IN ASCII FORMAT *
$ NOTE * BCD IF PRINTING IN BCD FORMAT *
$ NOTE * * *
$ NOTE * RMTA = STATION CODE FOR ASCII PRINT *
$ NOTE * * *
$ NOTE * RMTB = STATION CODE FOR BCD PRINT *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.XPDM *
$ NOTE * * *
$ NOTE *****
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCU/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.XPDM
$ FILE BM,C1S,1R
$ PTED30.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTED30
$ EXECUTE DUMP
$ LIMITS ,85K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL GS,W,R,$UMCU/$FILU.GS
$ PRMFL YS,W,R,$UMCU/$FILU.YS
$ FILE ED,D1S,10R
$ FILE TG,G1S,10R
$ FILE MB,C1R
$ SYSOUT EI,ORG
$ SYSOUT DD,ORG
$ IF 20,ERROR
$ PTED60.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTED60
$ EXECUTE DUMP
$ LIMITS ,70K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL GS,W,R,$UMCU/$FILU.GS
$ PRMFL YS,W,R,$UMCU/$FILU.YS
$ FILE ED,D1R
$ FILE TG,G1R
$ FILE S1,,100R

```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
XPDM: MASTER OUTLINE
XPDM: EXECUTION JCL

3
2
4

```
$      SYSOUT  EI,ORG
$      FILE    GP,E1S,10L
$      IF      20,ERROR
$      GOTO    P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$      CONVER
$      LIMITS  ,,10K
$      FILE    IN,E1R
$      SYSOUT  OT,&RMTB
$      OUTPUT  GBCD,MEDIA/3
$      GOTO    END
$ PASCII.
$ ASCII-PRINT 132 CH.
$      CONVER
$      LIMITS  ,,10K
$      FILE    IN,E1R
$      SYSOUT  OT,&RMTA
$      OUTPUT  ASCII,MEDIA/7
$ END.
$      CONVER
$      DATA   IN
***** XPDM - NORMAL END OF RUN *****
$      SYSOUT  OT,ORG
$      OUTPUT  MEDIA/03
$ ERROR.
$      ENDJOB
```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION	PAGE	173
PAFX: GENERAL-PURPOSE EXTRACTOR		3
PAFX: INTRODUCTION		3
		1

3.3. PAFX: GENERAL-PURPOSE EXTRACTOR

3.3.1. PAFX: INTRODUCTION

PAFX: GENERALIZED EXTRACTOR

PRINCIPLE

This procedure corresponds to a JCL model which allows the execution of a generalized extraction program generated by the XPAF procedure.

The procedure calls the JCL of PAFX74 (object type extraction) or PAFX85 (run-unit type extraction) according to the COBOL version which generated the extraction.

The external code of the extraction must be added to the EXTRAC parameter.

3.3.2. PAFX: USER INPUT

USER INPUT

One '*' line per library and session to be read.

```
-----  
! POS.! LEN.! VALUE ! MEANING !  
!-----!  
! 2 ! 1 ! '*' ! Line code !  
! 3 ! 8 ! !uuuuuuu! User code !  
! 11 ! 8 ! !pppppppp! User password !  
! 19 ! 3 ! !bbb ! Library code !  
! 22 ! 4 ! !nnnn ! Session number !  
! 26 ! 1 ! !T ! Session version !  
! 27 ! 1 ! !o ! Sub-network selected !  
-----
```

One command line 'X' for the following elements:

```
-----  
! POS.! LEN.! VALUE ! MEANING !  
!-----!  
! 2 ! 1 ! 'X' ! Line code !  
! 3 ! 4 ! !tttt ! Input type !  
! 7 ! 8 ! !ccccccc! Input code !  
! 15 ! 8 ! !bbbbbbbbb! Beginning delimiter !  
! 23 ! 8 ! !fffffff! End delimiter !  
! 31 ! 1 ! !'1' ! Debug option !  
! 32 ! 6 ! !xxxxxxx ! Maximum number of records in the !  
! ! ! ! ! temporary file !  
-----
```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
PAFX: GENERAL-PURPOSE EXTRACTOR
PAFX: DESCRIPTION OF STEPS, JCL

PAGE

175

3
3
3

3.3.3. PAFX: DESCRIPTION OF STEPS, JCL

DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

EXTRACTION: EXTRAC

The extraction JCL is called by a SELECT:

PRMFL: \$UMCU/\$JCL.PAFX74 if an object
PRMFL: \$UMCU/\$JCL.PAFX85 if a run-unit

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION

PAFX: GENERAL-PURPOSE EXTRACTOR

3

3

PAFX: DESCRIPTION OF STEPS, JCL

3

```

$      IDENT  $IDENT,$DEST.PAFX
$      NOTE   *****
$      NOTE   * VisualAge Pacbase
$      NOTE   * =====
$      NOTE   *
$      NOTE   *           EXAMPLE OF A USER EXTRACTION PROGRAM
$      NOTE   *
$      NOTE   * ENTER USER INPUT IN
$      NOTE   * $UMCU/$MB.PAFX
$      NOTE   *
$      NOTE   * *****
$      GLOBAL GDP=$GDP,USER=$USER,EXTRAC=CPGM
$ PTU001.
$      OPTION CBL74
$      SELECT $UMCS/$OBJBT.PTU001
$      EXECUTE DUMP
$      LIMITS ,13K
$      PRMFL  MB,R,S,$UMCU/$MB.PAFX
$      FILE   BM,C1S,1R
$ EXTRAC.
$      SELECT $UMCU/$JCL.PAFX&GDP
$      IF     20,ERROR
$ END.
$      CONVER
$      DATA  IN
***** PAFX - NORMAL END OF RUN *****
$      SYSOUT OT,ORG
$      OUTPUT MEDIA/03
$ ERROR.
$      ENDJOB

```


PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
PAFX: GENERAL-PURPOSE EXTRACTOR
PAFX: DESCRIPTION OF STEPS, JCL

PAGE

177

3
3
3

DESCRIPTION OF STEPS FOR COBOL-74

.Input files:

-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AR, BR
-Error message files
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AR, BR
-User input
File MB

.Input-Output files:

-PAF-extractor temporary files
File PA, YA

.Output files:

-Resulting unformatted extraction
PRMFL: \$UMCU/\$FILX/\$EXT.SO&USER OS
-Resulting formatted extraction
PRMFL: \$UMCU/\$FILX/\$EXT.SQ&USER QS

.Output report(s):

-Extraction report
SYSOUT DB

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
 PAFX: GENERAL-PURPOSE EXTRACTOR
 PAFX: DESCRIPTION OF STEPS, JCL

3
 3
 3

```

$      OPTION  CBL74
$      LIBRARY LA, LB
$      SELECT  $UMCU/$FILX/$O74.&EXTRAC
$      EXECUTE DUMP
$      LIMITS  20,150K
$      PRMFL   1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL   LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$      PRMFL   LA,R/C,R,$UMCS/$FILS.OBJLIB
$      PRMFL   AR,Q,R,$UMCB/$BASE.AR
$      PRMFL   BR,Q,R,$UMCB/$BASE.BR
$      PRMFL   AN,Q,R,$UMCB/$BASE.AN
$      PRMFL   BN,Q,R,$UMCB/$BASE.BN
$      PRMFL   AE,Q,R,$UMCB/$BASE.AE
$      PRMFL   XE,Q,R,$UMCB/$BASE.XE
$      FILE    PA,,100R
$      FILE    YA,,10R
$      FILE    MB,C1R
$      PRMFL   OS,W,S,$UMCU/$FILX/$EXT.SO&USER
$      PRMFL   QS,W,S,$UMCU/$FILX/$EXT.SQ&USER
$      FILE    S1,,20R
$      SYSOUT  DB,ORG
$      DATA   .U
FILE    FC/AN/,NBUF/16/,BFSZ/4096/
FILE    FC/BN/,NBUF/16/,BFSZ/4096/
FILE    FC/AR/,NBUF/16/,BFSZ/4096/
FILE    FC/BR/,NBUF/16/,BFSZ/4096/

```

DESCRIPTION OF STEPS FOR COBOL-85

.Input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
\$UMCU/\$BASE.BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AR, BR
-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AR, BR
-User input
File MB
-Instructions for dynamic data calls
DATA AA
-Allocation of buffers for PA, YA
DATA UF
.Input-output files:
-PAF-extractor temporary files
File PA, YA
.Output files:
-Resulting unformatted extraction
PRMFL: \$UMCU/\$FILX/\$EXT.SO&USER OS
-Resulting formatted extraction
PRMFL: \$UMCU/\$FILX/\$EXT.SQ&USER QS
.Output report(s):
-Extraction report
SYSOUT DB

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
 PAFX: GENERAL-PURPOSE EXTRACTOR
 PAFX: DESCRIPTION OF STEPS, JCL

3
 3
 3

```

$      RUN      RUFILX=$UMCU/$FILX/$RUNX.&EXTRAC,DIRFC=AA
$      DBASE    PACBASE($UMCS/$SCHEMA.LSTAR),
$      ETC      SSPB($UMCS/$SCHEMA.SSPB)
$      DATA    AA
ADD_RU $UMCS/$RUNS.PBBTST
ADD_RU $UMCS/$RUNS.PACA90
ADD_RU $UMCS/$RUNS.PACABE
ADD_RU $UMCS/$RUNS.SPABPB
$      DATA    Uf
FC/PA/ NBUFF/10/
FC/YA/ NBUFF/10/
$      LIMITS   20
$      PRMFL    AR,Q,R,$UMCB/$BASE.AR
$      PRMFL    BR,Q,R,$UMCB/$BASE.BR
$      PRMFL    AN,Q,R,$UMCB/$BASE.AN
$      PRMFL    BN,Q,R,$UMCB/$BASE.BN
$      PRMFL    AE,Q,R,$UMCB/$BASE.AE
$      PRMFL    XE,Q,R,$UMCB/$BASE.XE
$      FILE     PA,,100R
$      FILE     YA,,10R
$      FILE     MB,C1R
$      PRMFL    OS,W,S,$UMCU/$FILX/$EXT.SO&USER
$      PRMFL    QS,W,S,$UMCU/$FILX/$EXT.SQ&USER
$      FILE     S1,,20R
$      SYSOUT   DB,ORG
  
```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION	PAGE	181
PRGS: PRINTING OF MASTER PATH / OUTLINE FILE		3
PRGS: INTRODUCTION		4
		1

3.4. PRGS: PRINTING OF MASTER PATH / OUTLINE FILE

3.4.1. 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.

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
PRGS: PRINTING OF MASTER PATH / OUTLINE FILE
PRGS: USER INPUT

PAGE

182

3
4
2

3.4.2. 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 !  
-----
```

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
PRGS: PRINTING OF MASTER PATH / OUTLINE FILE
PRGS: DESCRIPTION OF STEPS

PAGE

183

3
4
3

3.4.3. PRGS: DESCRIPTION OF STEPS

PRGS: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

PRINTING OF THE MASTER PATH AND OUTLINE FILE:

.Input files:
-Error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Extraction paths
PRMFL : \$UMCU/\$FILU.GS \$UMCU/\$FILU.YS GS, YS

.Input transaction file:
-User input
File MB

.Output report:
-Execution report
SYSOUT DD
-PAC7GS report
SYSOUT GP

.Sort file(s):
File S1

PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
 PRGS: PRINTING OF MASTER PATH / OUTLINE FILE
 PRGS: EXECUTION JCL

3
 4
 4

3.4.4. PRGS: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.PRGS
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * PRINTING OF MASTER PATH AND MASTER OUTLINE FILE *
$ NOTE * * *
$ NOTE * PARAMETERS *
$ NOTE * * *
$ NOTE * IMP = ASCII FOR ASCII-FORMAT PRINTOUT *
$ NOTE * BCD FOR BCD-FORMAT PRINTOUT *
$ NOTE * * *
$ NOTE * RMTA = STATION CODE FOR ASCII PRINTOUT *
$ NOTE * * *
$ NOTE * RMTB = STATION CODE FOR BCD PRINTOUT *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.PRGS *
$ NOTE * * *
$ NOTE *****
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.PRGS
$ FILE BM,C1S,1R
$ PTEP90.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTEP90
$ EXECUTE DUMP
$ LIMITS ,85K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL GS,Q,R,$UMCU/$FILU.GS
$ PRMFL YS,Q,R,$UMCU/$FILU.YS
$ FILE GP,G1S,10L
$ FILE MB,C1R
$ FILE S1,,10R
$ SYSOUT EI,ORG
$ SYSOUT DD,ORG
$ IF 20,ERROR
$ GOTO P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$ CONVER
$ LIMITS ,,10K
$ FILE IN,G1R
$ SYSOUT OT,&RMTB
$ OUTPUT GBCD,MEDIA/3
$ GOTO END
$ PASCII.
$ ASCII-PRINT 132 CH.
$ CONVER
$ LIMITS ,,10K
$ FILE IN,G1R
$ SYSOUT OT,&RMTA
$ OUTPUT ASCII,MEDIA/7
$ END.
$ CONVER
$ DATA IN
***** PRGS - NORMAL END OF RUN *****

```


PERSONALIZED EXTRACTION & AUTOMATED DOCUMENTATION
PRGS: PRINTING OF MASTER PATH / OUTLINE FILE
PRGS: EXECUTION JCL

PAGE

185

3
4
4

\$ SYSOUT OT,ORG
\$ OUTPUT MEDIA/03
\$ ERROR.
\$ ENDJOB

VisualAge Pacbase - Operations Manual
BATCH PROCEDURES: USER'S GUIDE
QUALITY ANALYSIS AND CONTROL

PAGE 186

4

4. QUALITY ANALYSIS AND CONTROL

QUALITY ANALYSIS AND CONTROL	PAGE	187
ACTI: JOURNAL STATISTICS UTILITY		4
ACTI: INTRODUCTION		1

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 optional. The LINE, COLUMN, ABSCISSA, and ORDINATE lines are either required or prohibited, depending on the requested output report type.

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

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

Request lines must be entered in the following order:

OUTPUT PAGE AREA LINE COLUMN ABSCISSA ORDINATE

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

The user may enter up to 10 requests.

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: DEFINITION 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).

QUALITY ANALYSIS AND CONTROL
ACTI: JOURNAL STATISTICS UTILITY
ACTI: COMMAND LANGUAGE

PAGE

190

4
1
2

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.

QUALITY ANALYSIS AND CONTROL
ACTI: JOURNAL STATISTICS UTILITY
ACTI: COMMAND LANGUAGE

4
1
2

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.

QUALITY ANALYSIS AND CONTROL
 ACTI: JOURNAL STATISTICS UTILITY
 ACTI: COMMAND LANGUAGE

4
 1
 2

&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).

QUALITY ANALYSIS AND CONTROL
 ACTI: JOURNAL STATISTICS UTILITY
 ACTI: COMMAND LANGUAGE

4
 1
 2

&MON MONTH-BY-MONTH PRESENTATION

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

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

&YEAR YEAR-BY-YEAR PRESENTATION

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

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

&SESS PRESENTATION BY SESSION

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

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

&CHAR PRINTING CURVE CHARACTER

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

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

&INTR NUMBER OF INPUT TRANSACTIONS

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

&RETR NUMBER OF REAL TRANSACTIONS

May only be used to define the activity calculation mode.

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

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

= : 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 messages, 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.

QUALITY ANALYSIS AND CONTROL
ACTI: JOURNAL STATISTICS UTILITY
ACTI: USER INPUT

PAGE

200

4
1
3

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

TRANSACTION RECOGNITION: PTU001

EXTRACTION: PTU630

.Permanent input files:
-Error message file
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Journal Backup File
 PRMFL : \$UMCU/\$FILU.ARCH(0) PJ

.Transaction file:
-Update transactions
 File MB

.Output file
-Transactions for selected reports
 File (FLR 200, CISZ 2048) ST

.Output report:
-Batch-procedure authorization option
 SYSOUT DD

.Return code(s):
-Switch-20
 1 - No authorization on the batch procedure

PRINTING OF RESULTS: PTU640

.Permanent input file:
-Error Messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Input file:
-Transactions for selected reports
 File TS

.Output report:
-Selected reports
 SYSOUT IV

.Sort file(s):
 File S1

QUALITY ANALYSIS AND CONTROL
 ACTI: JOURNAL STATISTICS UTILITY
 ACTI: EXECUTION JCL

4
 1
 5

4.1.5. ACTI: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.ACTI
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * JOURNAL STATISTICS *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.ACTI *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.PJ0
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.ACTI
$ FILE BM,C1S,1R
$ PTU630.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTU630
$ EXECUTE DUMP
$ LIMITS ,90K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,R/C,R,$UMCB/$BASE.AE
$ PRMFL XE,R/C,R,$UMCB/$BASE.XE
$ PRMFL PJ,R,R,&PJI
$ FILE MB,C1R
$ SYSOUT DD,ORG
$ SYSOUT EI,ORG
$ FILE ST,M1S,10R
$ IF 20,ERROR
$ PTU640.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTU640
$ EXECUTE DUMP
$ LIMITS ,75K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE TS,M1R
$ FILE S1,,10R
$ SYSOUT EI,ORG
$ SYSOUT IV,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** NORMAL END OF RUN = ACTI *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

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: QUALITY CONTROL ANALYSIS

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 comprises several monitors (PACBA, PACBE, ...) which ensure the linking of the various sub-programs associated to one type of generation.

Its operation is identical to that of the standard GPRT generation-print procedure. For further details on its operation, you should therefore refer to the chapter dedicated to GPRT.

The PACBx monitors are those of GPRT. The PACQ monitor is made up of the programs specific to PQC: PTUQ20, PTUQ30, PTUQ40, and PTUQ50.

The first monitor (PACBA) interprets user requests and stores them in the production environment if necessary.
Then, the other monitors are activated if necessary, in the following order:

- Screens
- Batch programs
- Client part of Client/Server applications
- Server part of Client/Server applications

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 PTUQ30, PTUQ40 and PTUQ50 programs.

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

EXECUTION CONDITIONS

The files may remain open, except when the user required the on-line requests to be processed, via the '+AG' command. In this case, the generation/print request file must be closed.

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

TRANSACTION RECOGNITION: PTU001

GENERATION-PRINTING: PACBx

Please refer to the chapter dedicated to the GPRT procedure.

QUALITY ANALYSIS: PACQ

This program is a monitor grouping the following sub-programs:

PTUQ20, PTUQ30, PTUQ40, PTUQ50.

.Permanent input files:

-Error message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Quality-rule file
PRMFL : \$UMCU/\$MV.PQCE MI

.Transaction file

-Selection parameter input
File MC

.Sort file:

File S1

.Outputreports:

-Selection-parameter check
SYSOUT IB
-Results by entity type
SYSOUT IE
-Results by entity
SYSOUT IF
-List of VisualAge Pacbase identifiers
higher than the indicators
SYSOUT IG

.Return code:

-Switch-29
1 - A minimum mark was required on an M-type line,
and it was not reached.

DOCUMENTATION PRINTING: PACBED

Please refer to the chapter dedicated to the GPRT procedure.

QUALITY ANALYSIS AND CONTROL
PQC-: PACBENCH QUALITY CONTROL
PQCA: EXECUTION JCL

4
2
4

4.2.4. PQCA: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.PQCA
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * PACBENCH QUALITY CONTROL *
$ NOTE * ANALYSIS *
$ NOTE * * *
$ NOTE * SYMBOLICS *
$ NOTE * * *
$ NOTE * USER = USER CODE FOR FILES SUFFIX *
$ NOTE * * *
$ NOTE * IMP = BCD IF PRINTING WITH BCD FORMAT *
$ NOTE * = ASCII IF PRINTING WITH ASCII FORMAT *
$ NOTE * * *
$ NOTE * ACCES = L IF REQUEST '+AG' *
$ NOTE * Q OTHERWISE *
$ NOTE * * *
$ NOTE * TP8 = O IF TP8 MONITOR *
$ NOTE * N IF DMIV-TP MONITOR *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.QCA1 *
$ NOTE * $UMCU/$MB.QCA2 *
$ NOTE * * *
$ NOTE * THE INPUT FILE MBQCA1 IS THE SAME AS GPRT INPUT *
$ NOTE * TRANSACTION. *
$ NOTE * * *
$ NOTE * THE INPUT FILE MQBCA2 CORRESPONDS TO THE *
$ NOTE * PARAMETERS FOR SELECTION OF THE QUALITY RULES. *
$ NOTE * * *
$ NOTE *****
$ GLOBAL USER=( $USER ),ACCES=Q,ESTEP=PACCQ
$ GLOBAL JCL=( $UMCU/$JCL ),TP8=$TP8,IMP=BCD
$ SELECT &JCL/PGPRT
$ INPUT1. ENTREES UTILISATEURS TYPE GPRT
$ SELECT &JCL/INPUT1
$ INPUT2. PARAMETRES DE SELECTION
$ SELECT &JCL/INPUT2
$ PACBA. ANALYSE DES DEMANDES
$ SELECT &JCL/PACBA
$ PACBE. GENERATION DIALOGUE
$ GLOBAL MONI=PACBE
$ GLOBAL TYPP=GE
$ SELECT &JCL/DRVR
$ SELECT &JCL/PACBE
$ SELECT &JCL/PAF
$ SELECT &JCL/EMONI
$ PACBP. GENERATION BATCH
$ GLOBAL MONI=PACBP
$ GLOBAL TYPP=GP
$ SELECT &JCL/DRVR
$ SELECT &JCL/PACBP
$ SELECT &JCL/PAF
$ SELECT &JCL/EMONI
$ PACBG. GENERATION CLIENT
$ GLOBAL MONI=PACBG
$ GLOBAL TYPP=GG
$ SELECT &JCL/DRVR
$ SELECT &JCL/PACBG
$ SELECT &JCL/PAF
$ SELECT &JCL/EMONI
$ PACBV. GENERATION SERVEUR
$ GLOBAL MONI=PACBV
$ GLOBAL TYPP=GV
$ SELECT &JCL/DRVR
$ SELECT &JCL/PACBV
$ SELECT &JCL/PAF
$ SELECT &JCL/EMONI

```

QUALITY ANALYSIS AND CONTROL
PQC-: PACBENCH QUALITY CONTROL
PQCA: EXECUTION JCL

4
2
4

\$ PACCQ. CONTROLE QUALITE
\$ SELECT &JCL/PACCQ
\$ PACBED. EDITIONS
\$ SELECT &JCL/PACBED

QUALITY ANALYSIS AND CONTROL

PQC-: PACBENCH QUALITY CONTROL

PQCA: EXECUTION JCL

4
2
4

```

$      NOTE *****
$      NOTE * VisualAge Pacbase *
$      NOTE * ===== *
$      NOTE * * *
$      NOTE * PACBENCH QUALITY CONTROL: PROCEDURE CALLED BY PQCA *
$      NOTE * * *
$      NOTE *****
$ PACCQ.
$      DEFAULT LTCPCQ=20,LTPGCQ=200K,LTOUCQ=30K
$      PROGRAM RLHS,ON1,DUMP
$      LIMITS &LTCPCQ,&LTPGCQ
$      PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL AE,Q,R,$UMCB/$BASE.AE
$      PRMFL XE,Q,R,$UMCB/$BASE.XE
$      PRMFL MI,R,R,$UMCU/$MV.PQCE
$      FILE MC,C2
$      FILE MJ,,50R
$      FILE MK,,50R
$      FILE MM,,50R
$      FILE MN,,50R
$      FILE MO,,50R
$      FILE MZ,,50R
$      FILE KD,K2S
$      FILE KE,K3S
$      FILE KG,V1S
$      FILE KP,K7S
$      FILE KU,K9S
$      FILE KV,V2S
$      FILE YL,Y1SS
$      FILE YM,Y2SS
$      FILE S1,,100R
$      FILE IB,E2S,50L
$      FILE IE,E3S,50L
$      FILE IF,E4S,50L
$      FILE IG,E5S,50L
$      SYSOUT EI,ORG
$      PRMFL H*,R/C,R,$UMCS/$HSTAR.PACQ
$      IF 20,ERROR
$      GOTO CQ&IMP
$ CQASCII.
$      CONVER
$      LIMITS ,,,&LTOUCQ
$      FILE IN,E2R
$      FILE ",E3R
$      FILE ",E4R
$      FILE ",E5R
$      SYSOUT OT,&RMTA
$      OUTPUT ASCII,MEDIA/7
$      GOTO CQEND
$ CQBCD.
$      CONVER
$      LIMITS ,,,&LTOUCQ
$      FILE IN,E2R
$      FILE ",E3R
$      FILE ",E4R
$      FILE ",E5R
$      SYSOUT OT,&RMTB
$      OUTPUT GB CD,MEDIA/3
$ CQEND.

```

QUALITY ANALYSIS AND CONTROL	PAGE	210
PQC-: PACBENCH QUALITY CONTROL		4
PQCE: EXTRACTION OF USER-DEFINED QUALITY RULES		2
		5

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 ! WLEX ! 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.

QUALITY ANALYSIS AND CONTROL
PQC- : PACBENCH QUALITY CONTROL
PQCE: USER INPUT

PAGE

212

4
2
6

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

TRANSACTION RECOGNITION: PTU001

EXTRACTION: PACX

This step extracts transactions according to user input.

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Erroneous transactions
PRMFL : \$UMCU/\$FILU.ARCH(0) PJ

.Input transaction file:
-User input
File MB

.Work files:
-User input
File (FLR 80, CISZ 512) BM
-EXPU work file
File (FLR 55, CISZ 6144) MM
-EXPJ work file
File (FLR 152, CISZ 2816) MJ
-RMEN work file
File (FLR 180, CISZ 4608) TE
-RMEN work file
File (FLR 12, CISZ 4088) RE
-RMEN work file
File (FLR 167, CISZ 9413) RM

-Extracted transactions
File (FLR 167, CISZ 9413) WD
-Multi-layered Extractor work file
File (FLR 112, CISZ 3072, UIND) XX, XY

QUALITY ANALYSIS AND CONTROL

PQC-: PACBENCH QUALITY CONTROL

PQCE: DESCRIPTION OF STEPS

4
2
7

.Output files:

-Extracted transactions for UPDT
 PRMFL : \$UMCU/\$MV.UPDT MV

-Extracted transactions for REOR (EXPU)
 PRMFL : \$UMCU/\$MV.REOR MR

-Extracted transactions for UPDP
 PRMFL : \$UMCU/\$MV.UPDP GY

-Extracted transactions for CPSN
 PRMFL : \$UMCU/\$FILU.MACPSN (ou SLCPSN) TD

-Extracted transactions for EXUE
 PRMFL : \$UMCU/\$MV.EXUE UE

.Output reports:

-General printout of the program stream
 SYSOUT IA

-List of errors on input transactions
 SYSOUT DD

-Summary reports on extractions

SYSOUT EE
 SYSOUT EP
 SYSOUT EQ
 SYSOUT EZ

.Sort file(s):

File S1, S2, S3, S4

.Return codes:

0 - No error

4 - Error in user input (specified in EE)
 or
 EXTR/EXUE - problem during extraction
 (specified in EZ)

8 - Error on '*'-line (specified in DD)
 or
 EXLI - Database unavailable

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
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

-Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR

QUALITY ANALYSIS AND CONTROL

PQC-: PACBENCH QUALITY CONTROL

PQCE: DESCRIPTION OF STEPS

4
2
7

.Permanent output file:
 -'Compiled' Quality Rules
 PRMFL : \$UMCU/\$MV.PQCE MI

.Transaction files:
 -User input
 File MB
 -User entity occurrences
 File MC

.Output file:
 -Preparation for printing
 File ML

.Output report(s):
 -Rule-validity report
 SYSOUT IC
 -Batch-procedure authorization option
 SYSOUT DD

.Sort file(s):
 File S1

PRINTING OF QUALITY RULES: PTUQ15

.Permanent input file:
 -Error message file
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Input file:
 -Preparation for printing
 File MT

.Output reports:
 -List of quality factors and criteria,
 and description by indicator
 SYSOUT II
 -Dictionary of Quality rules
 SYSOUT IJ

.Sort file(s):
 File S1

QUALITY ANALYSIS AND CONTROL

PQC- : PACBENCH QUALITY CONTROL

PQCE: EXECUTION JCL

4

2

8

4.2.8. PQCE: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.PQCE
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * *
$ NOTE * PACBENCH QUALITY CONTROL *
$ NOTE * EXTRACTION *
$ NOTE * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.PQCE *
$ NOTE * *
$ NOTE * THE STRUCTURE OF THE INPUT TRANSACTIONS IS THE *
$ NOTE * SAME AS THE PACX INPUT TRANSACTIONS WITH *
$ NOTE * EXTRACTOR CODE = EXUE *
$ NOTE * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.PQCE
$ FILE BM,C1S,1R
$ EXUE.
$ RUN RUFIL=$UMCS/$RUNS.PACX,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ DATA AA
ADD_RU $UMCS/$RUNS.PACA90
ADD_RU $UMCS/$RUNS.PACABE
ADD_RU $UMCS/$RUNS.PACCTL
ADD_RU $UMCS/$RUNS.PACHOI
ADD_RU $UMCS/$RUNS.PACS30
ADD_RU $UMCS/$RUNS.PACS75
ADD_RU $UMCS/$RUNS.SPABPB
$ DATA UE
FC/XX/ NBUFF/10/
FC/XY/ NBUFF/10/
$ LIMITS 20
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE MB,C1S
$ FILE UE,C2S,10L
$ FILE BM,,10R
$ FILE WD,,10R
$ FILE XX,,100R
$ FILE XY,,10R
$ FILE S1,,50R
$ SYSOUT DD,ORG
$ SYSOUT EE,ORG
$ SYSOUT EI,ORG
$ SYSOUT EZ,ORG
$ SYSOUT IA,ORG
$ IF 20,ERROR
$ PTUQ10.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PTUQ10
$ EXECUTE DUMP
$ LIMITS ,70K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE

```


QUALITY ANALYSIS AND CONTROL

PQC- : PACBENCH QUALITY CONTROL

PQCE: EXECUTION JCL

4

2

8

```
$      PRMFL  AR,Q,R,$UMCB/$BASE.AR
$      PRMFL  BR,Q,R,$UMCB/$BASE.BR
$      PRMFL  MI,W,R,$UMCU/$MV.PQCE
$      FILE   MB,C1
$      FILE   MC,C2
$      FILE   ML,C3S,50R
$      FILE   S1,,100R
$      SYSOUT EI,ORG
$      SYSOUT IC,ORG
$      SYSOUT DD,ORG
$      IF     20,ERROR
$ PTUQ15.
$      OPTION CBL74
$      LIBRARY LA,LB
$      SELECT $UMCS/$OBJBT.PTUQ15
$      EXECUTE DUMP
$      LIMITS ,75K
$      PRMFL  1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL  LA,R/C,R,$UMCS/$FILS.OBJLIB
$      PRMFL  LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$      PRMFL  AE,Q,R,$UMCB/$BASE.AE
$      PRMFL  XE,Q,R,$UMCB/$BASE.XE
$      FILE   MT,C3
$      FILE   S1,,100R
$      SYSOUT II,ORG
$      SYSOUT IJ,ORG
$      SYSOUT EI,ORG
$      IF     20,ERROR
$ END.
$      CONVER
$      DATA  IN
***** PQCE - NORMAL END OF RUN *****
$      SYSOUT OT,ORG
$      OUTPUT MEDIA/03
$ ERROR.
$      ENDJOB
```

VisualAge Pacbase - Operations Manual
BATCH PROCEDURES: USER'S GUIDE
METHODOLOGY INTEGRITY CHECK

PAGE 218

5

5. METHODOLOGY INTEGRITY CHECK

METHODOLOGY INTEGRITY CHECK	PAGE	219
ADM: SSADM PACDESIGN METHODOLOGY		5
SADM: INTRODUCTION		1
		1

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 ! eeeee ! 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.

METHODOLOGY INTEGRITY CHECK
ADM: SSADM PACDESIGN METHODOLOGY
SADM: DESCRIPTION OF STEPS

PAGE

221

5
1
3

5.1.3. SADM: DESCRIPTION OF STEPS

SADM: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

SSADM-ENTITY CONSISTENCY CHECK: PADM10

.Permanent input files:
-Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
 PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error-message file
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Transaction file:
-User input
 File MB

.Work file(s):
-Fichier indexé standard PAF
 File PA, YA

.Output report:
-List of checked SSADM entities
 SYSOUT EJ

METHODOLOGY INTEGRITY CHECK
 ADM: SSADM PACDESIGN METHODOLOGY
 SADM: EXECUTION JCL

5
 1
 4

5.1.4. SADM: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.SADM
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * PACDESIGN SSADM *
$ NOTE * * *
$ NOTE * INTEGRITY CHECKING *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.SADM *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.SADM
$ FILE BM,C1S,1R
$ PADM10.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PADM10
$ EXECUTE DUMP
$ LIMITS ,180K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE MB,C1
$ FILE PA,,50R
$ FILE YA,,10R
$ SYSOUT AJ,ORG
$ END.
$ CONVER
$ DATA IN
***** SADM - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

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	! eeeee	! Entity code (PRC)

METHODOLOGY INTEGRITY CHECK

YSM: WORKSTATION / YSM METHODOLOGY

YSMC: USER INPUT

5

2

2

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  ! eeeee ! 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

TRANSACTION RECOGNITION: PTU001

YSM METHOD INTEGRITY CHECKING: PYSMCC

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Transaction file:
-User input
File MB

.Work file(s):
-PAF standard indexed file
File PA, YA

.Output reports:
-Integrity checking lists
SYSOUT EJ
-Validation reports
SYSOUT EI

INTER-PROCESS CONSISTENCY: PYSMC3

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Transaction file:
-User input
File MB

METHODOLOGY INTEGRITY CHECK
YSM: WORKSTATION / YSM METHODOLOGY
YSMC: DESCRIPTION OF STEPS

PAGE

227

5
2
3

.Work file(s):
-PAF standard indexed file
File PA, YA

.Output report:
-Integrity-check lists
SYSOUT EJ

LIST OF RELATIONSHIPS AND REPORTS: PYSMC2

.Permanent input files:
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Error messages
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Transaction file:
-User input
File MB

.Work file(s):
-PAF standard indexed file
File PA, YA

.Output report:
-Integrity-check lists
SYSOUT EJ

METHODOLOGY INTEGRITY CHECK

YSM: WORKSTATION / YSM METHODOLOGY

YSMC: EXECUTION JCL

5

2

4

5.2.4. YSMC: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.YSMC
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * PACDESIGN YSM *
$ NOTE * * *
$ NOTE * INTEGRITY CHECKING *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.YSMC *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.YSMC
$ FILE BM,C1S,1R
$ PYSMCC.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PYSMCC
$ EXECUTE DUMP
$ LIMITS 20,180K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE MB,C1S
$ FILE PA,,50R
$ FILE YA,,10R
$ SYSOUT EI,ORG
$ SYSOUT AI,ORG
$ SYSOUT AJ,ORG
$ IF 20,ERROR
$ PYSMC3.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PYSMC3
$ EXECUTE DUMP
$ LIMITS 20,180K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE MB,C1S
$ FILE PA,,50R
$ FILE YA,,10R
$ SYSOUT EI,ORG
$ SYSOUT AJ,ORG
$ IF 20,ERROR
$ PYSMC2.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PYSMC2

```

METHODOLOGY INTEGRITY CHECK

YSM: WORKSTATION / YSM METHODOLOGY

YSMC: EXECUTION JCL

5

2

4

```
$ EXECUTE DUMP
$ LIMITS 20,180K,,10K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE MB,C1
$ FILE PA,,50R
$ FILE YA,,10R
$ SYSOUT EI,ORG
$ SYSOUT AJ,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** YSMC - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB
```

VisualAge Pacbase - Operations Manual
BATCH PROCEDURES: USER'S GUIDE
PACTABLES

PAGE 230

6

6. PACTABLES

	PAGE	231
PACTABLES		6
GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR		1
GETD - GETA: INTRODUCTION		1

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.

PACTABLES	
GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR	
GETD - GETA: INTRODUCTION	

6
1
1

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.

GCOS8 PLATFORM SPECIFIC:

The interface management is ensured by the GETD procedure alone.

It operates with a sequential indexed file UFAS, which is an image of the Pactables table-descriptions area.

This file is initialized by the GETI procedure, or by the LOTD Pactables procedure, depending on the context. (See the general introduction.)

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR

GETD - GETA: USER INPUT

6

1

2

6.1.2. GETD - GETA: USER INPUT

GETD-GETA: USER INPUTUSER INPUT

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

```

-----!
!POS.!LEN.! VALUE ! MEANING !
!-----!
! 2 ! 1 ! '*' ! Line code !
! 3 ! 8 ! !uuuuuuu! User code !
! 11 ! 8 ! !pppppppp! User password !
! 19 ! 3 ! !bbb ! Library code !
! 22 ! 4 ! !nnnn ! Session number !
! 26 ! 1 ! !t ! Session status !
!-----!

```

One 'Z' line per generation or print request.

```

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

```

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR
 GETD - GETA: USER INPUT

6
 1
 2

```

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

```

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

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

RESULT OBTAINED

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

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR

6

GETD - GETA: DESCRIPTION OF STEPS

1

3

6.1.3. GETD - GETA: DESCRIPTION OF STEPS

GETD: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001EXTRACTION & UPDATE PREPARATION: PACT40

.Permanent input files:

- VisualAge Pacbase data file
 - PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
- VisualAge Pacbase index file
 - PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
- VisualAge Pacbase error-message file
 - PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
- Table-description file
 - PRMFL : \$UMCU/\$FILU.TD \$UMCU/\$FILU.YD TD, YD

.Input transaction file:

- User requests
 - File MB

.Output report(s):

- Transaction summary
 - SYSOUT ET
- Batch-procedure authorization option
 - SYSOUT DD

.Output file:

- Descriptions update transactions higher or equal to 2.0
 - PRMFL : \$UMCU/\$MV.GETT MD

.Return code:

- Switch-20
 - 1 - No authorization on the batch procedure

FORMATTING OF DESCRIPTIONS < R 2.0: PACT45

.Input file:

- Description-update transactions higher or equal to 2.0
 - File MD

.Output file

- Description-update transactions lower or equal to 2.0
 - PRMFL : \$UMCU/\$MV.GETT12 ND

UPDATE OF TABLE-DESCRIPTION FILE: PACT50

.Permanent input file:

- Table-description file
 - PRMFL : \$UMCU/\$FILU.TD \$UMCU/\$FILU.YD TD, YD

.Input transaction files:

- User requests
 - File MB
- Update transactions
 - File MD

.Output report:

- Update review
 - SYSOUT ET

.Sort file(s):

- File S1, S2

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR
 GETD: EXECUTION JCL

6
 1
 4

6.1.4. GETD: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.GETD
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * TABLE DESCRIPTION GENERATION *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.GETA *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.GETA
$ FILE BM,C1S,1R
$ PACT40.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PACT40
$ EXECUTE DUMP
$ LIMITS ,75K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL TD,Q,R,$UMCU/$FILU.TD
$ PRMFL YD,Q,R,$UMCU/$FILU.YD
$ FILE MB,C1R
$ PRMFL MD/D1S,L,R,$UMCU/$MV.GETT
$ SYSOUT EI,ORG
$ SYSOUT DD,ORG
$ SYSOUT ET,ORG
$ IF 20,ERROR
$ PACT45.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PACT45
$ EXECUTE DUMP
$ LIMITS ,35K
$ FILE MD,D1S
$ PRMFL ND,W,R,$UMCU/$MV.GETT12
$ PACT50.
$ OPTION CBL74
$ LIBRARY LA
$ SELECT $UMCS/$OBJBT.PACT50
$ EXECUTE DUMP
$ LIMITS ,35K
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL TD,L,R,$UMCU/$FILU.TD
$ PRMFL YD,L,R,$UMCU/$FILU.YD
$ FILE MD,D1R
$ FILE S1,,20R
$ SYSOUT ET,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** GETD - NORMAL END OF RUN *****
$ SYSOUT OT,ORG

```

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR

GETD: EXECUTION JCL

6

1

4

\$ OUTPUT MEDIA/03

\$ ERROR.

\$ ENDJOB

	PAGE	238
PACTABLES		6
GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR		1
GET2 - GET1: INTRODUCTION		5

6.1.5. 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.)

	PAGE	239
PACTABLES		6
GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR		1
GET2 - GET1: INTRODUCTION		5

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 abnormally while the Table-Description file is being updated, the file must be restored before the procedure can be restarted.

GCOS8 PLATFORM SPECIFIC:

The interface management is ensured by the GET2 procedure alone.

It operates with a sequential indexed file UFAS, which is an image of the Pactables table-descriptions area.

This file is initialized by the GET0 procedure, or by the LOTD Pactables procedure, depending on the context. (See the general introduction.)

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR

GET2 - GET1: USER INPUT

6

1

6

6.1.6. GET2 - GET1: USER INPUT

GET2-GET1: USER INPUTUSER INPUT

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

```

-----
!Pos.! Len.! Value  ! Meaning
-----
!  2 !   1 ! '*'    ! Line code
!  3 !   8 ! uuuuuuu ! 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 ! DDDMMY ! Date from which the table description
!   !   !       ! can be modified (optional)
-----

```


PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR

GET2 - GET1: USER INPUT

6

1

6

```

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

```

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

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

RESULT

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

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR

GET2 - GET1: DESCRIPTION OF STEPS

PAGE

242

6

1

7

6.1.7. GET2 - GET1: DESCRIPTION OF STEPS

GET2: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

EXTRACTION AND UPDATE PREPARATION: PACT41

.Permanent input files:
-VisualAge Pacbase Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-VisualAge Pacbase Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-VisualAge Pacbase Error-message file
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Table-description file
PRMFL : \$UMCU/\$FILU.TD \$UMCU/\$FILU.YD TD, YD

.Input Transaction file:
-Descriptions requests
File MB

.Output reports:
-Transaction report
SYSOUT ET
-Batch-procedure authorization option
SYSOUT DD

.Output file:
-2.0-Description update transactions
PRMFL : \$UMCU/\$MV.GETT MD

.Return code(s):
Switch-20
1 - No authorization on the Batch procedures

TABLE-DESCRIPTION UPDATE: PACT51

(GET2 procedure only)

.Permanent input file:
-Table-description file
PRMFL : \$UMCU/\$FILU.TD \$UMCU/\$FILU.YD TD, YD

.Input transaction files:
-Descriptions requests
File MB

-Update transactions
PRMFL : \$UMCU/\$MV.GETT MD

.Output report:
-Update report
SYSOUT ET

.Sort files:
File S1, S2

PACTABLES

GETD-GETA-GET1-GET2: DESCRIPTION GENERATOR
 GET2: EXECUTION JCL

6
 1
 8

6.1.8. GET2: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.GET2
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * TABLE DESCRIPTION GENERATION *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.GETA *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.GETA
$ FILE BM,C1S,1R
$ PACT41.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PACT41
$ EXECUTE DUMP
$ LIMITS ,75K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL TD,Q,R,$UMCU/$FILU.TD
$ PRMFL YD,Q,R,$UMCU/$FILU.YD
$ FILE MB,C1R
$ PRMFL MD/D1S,L,R,$UMCU/$MV.GETT
$ SYSOUT EI,ORG
$ SYSOUT DD,ORG
$ SYSOUT ET,ORG
$ IF 20,ERROR
$ PACT51.
$ OPTION CBL74
$ LIBRARY LA
$ SELECT $UMCS/$OBJBT.PACT51
$ EXECUTE DUMP
$ LIMITS ,35K
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL TD,L,R,$UMCU/$FILU.TD
$ PRMFL YD,L,R,$UMCU/$FILU.YD
$ FILE MD,D1R
$ FILE S1,,20R
$ SYSOUT ET,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** GET2 - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

PACTABLES	PAGE	244
GETI-GET0: INITIALIZATION OF DESCRIPTION FILE		6
GETI: INTRODUCTION		2
		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.

PACTABLES
GETI-GET0: INITIALIZATION OF DESCRIPTION FILE
GETI: DESCRIPTION OF STEPS

PAGE

245

6
2
2

6.2.2. GETI: DESCRIPTION OF STEPS

GETI: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

INITIALIZATION OF DESCRIPTION FILE: PACTIN

.Permanent output file:
-Table description file
PRMFL : \$UMCU/\$FILU.TD \$UMCU/\$FILU.YD TD, YD

.Transaction input file:
-Parameter line
File MB

```
+-----+-----+-----+-----+
!POS.!LEN. ! VALUE ! MEANING !
+-----+-----+-----+-----+
! 1 ! 36 !      ! Installation name !
! 37 ! 1 !      ! Language code:   !
!   !   ! 'F' ! French (Default option) !
!   !   ! 'E' ! English          !
+-----+-----+-----+-----+
! 51 ! 4 ! cccc ! Class for security system !
! 55 ! 1 !      ! Type of security system !
!   !   ! 'R' ! RACF                !
!   !   ! 'S' ! TOP SECRET         !
! 56 ! 2 ! nn   ! Number of lines per printing page !
! 58 ! 1 !      ! Type of resource controls !
!   !   ! ' ' ! Def.tables resources security system !
!   !   ! 'P' ! Def.resources in VA Pabase !
! 59 ! 1 !      ! Lock of the user's code !
!   !   ! ' ' ! Other user's code authorized !
!   !   ! 'N' ! Other user's code unauthorized !
+-----+-----+-----+-----+
```

Output report:
-Initialization review
SYSOUT ED

PACTABLES

GETI-GET0: INITIALIZATION OF DESCRIPTION FILE

GETI: EXECUTION JCL

6

2

3

6.2.3. GETI: EXECUTION JCL

```

$   IDENT   $IDENT,$DEST.GETI
$   NOTE    *****
$   NOTE    * VisualAge Pacbase                               *
$   NOTE    * =====                                         *
$   NOTE    *                                                                                             *
$   NOTE    *                               TABLE INITIALIZATION                               *
$   NOTE    *                                                                                             *
$   NOTE    *   ENTER USER INPUT IN                               *
$   NOTE    * $UMCU/$MB.GETI                                     *
$   NOTE    *                                                                                             *
$   NOTE    * *****
$ PTU001.
$   OPTION  CBL74
$   SELECT  $UMCS/$OBJBT.PTU001
$   EXECUTE DUMP
$   LIMITS  ,13K
$   PRMFL   MB,R,S,$UMCU/$MB.GETI
$   FILE    BM,C1S,1R
$ PACTIN.
$   OPTION  CBL74
$   LIBRARY LA
$   SELECT  $UMCS/$OBJBT.PACTIN
$   EXECUTE DUMP
$   LIMITS  ,30K
$   PRMFL   LA,R/C,R,$UMCS/$FILS.OBJLIB
$   PRMFL   TD,L,R,$UMCU/$FILU.TD
$   PRMFL   YD,L,R,$UMCU/$FILU.YD
$   FILE    MD,C1R
$   SYSOUT  EI,ORG
$   SYSOUT  ED,ORG
$   IF      20,ERROR
$ END.
$   CONVER
$   DATA   IN
***** GETI - NORMAL END OF RUN *****
$   SYSOUT  OT,ORG
$   OUTPUT  MEDIA/03
$ ERROR.
$   ENDJOB

```

PACTABLES	PAGE	247
GETI-GET0: INITIALIZATION OF DESCRIPTION FILE		6
GET0: INTRODUCTION		2
		4

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.

PACTABLES

GETI-GET0: INITIALIZATION OF DESCRIPTION FILE

GET0: DESCRIPTION OF STEPS

6

2

5

6.2.5. GET0: DESCRIPTION OF STEPS

GET0: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001INITIALIZATION OF DESCRIPTION FILE: PACTI1

.Permanent output file:
 -Table-description file
 PRMFL : \$UMCU/\$FILU.TD \$UMCU/\$FILU.YD TD, YD

.Input transaction file:
 -Parameter line
 File MB

```
-----
!Pos.! Len.! Value ! Meaning !
!-----!
! 1 ! 36 !           ! Installation label !
! 37 ! 1 !           ! Language code      !
!   !   ! 'F'       ! French (default option) !
!   !   ! 'E'       ! English             !
!-----!
```

.Output report:
 -Initialization report
 SYSOUT ED

PACTABLES

GETI-GET0: INITIALIZATION OF DESCRIPTION FILE

GET0: EXECUTION JCL

6

2

6

6.2.6. GET0: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.GET0
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * TABLE INITIALIZATION *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.GETI *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.GETI
$ FILE BM,C1S,1R
$ PACTI1.
$ OPTION CBL74
$ LIBRARY LA
$ SELECT $UMCS/$OBJBT.PACTI1
$ EXECUTE DUMP
$ LIMITS ,30K
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL TD,L,R,$UMCU/$FILU.TD
$ PRMFL YD,L,R,$UMCU/$FILU.YD
$ FILE MD,C1R
$ SYSOUT EI,ORG
$ SYSOUT ED,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** GET0 - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

VisualAge Pacbase - Operations Manual
BATCH PROCEDURES: USER'S GUIDE
PAC/IMPACT

PAGE 250

7

7. PAC/IMPACT

	PAGE	251
PAC/IMPACT	7	
FOREWORD	7	

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 performed 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.

PAC/IMPACT	PAGE	252
ISEP: SELECTION OF ENTRY POINTS		7
ISEP: INTRODUCTION		1

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).

	PAGE	253
PAC/IMPACT		7
ISEP: SELECTION OF ENTRY POINTS		1
ISEP: INTRODUCTION		1

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.

PAC/IMPACT

ISEP: SELECTION OF ENTRY POINTS

ISEP: USER INPUT

7

1

2

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 inc-
!   !   !             ! luded 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
-----

```

PAC/IMPACT

ISEP: SELECTION OF ENTRY POINTS

ISEP: USER INPUT

7

1

2

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
-----

```

PAC/IMPACT

ISEP: SELECTION OF ENTRY POINTS

7

ISEP: DESCRIPTION OF STEPS

1

3

7.1.3. ISEP: DESCRIPTION OF STEPS

ISEP: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001SELECTION OF ENTRY POINTS: PAN210

.Permanent input files:

- Error messages
 - PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
- Data file
 - PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
- Index file
 - PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
- File of entities to be analyzed
 - PRMFL : \$UMCU/\$FILU.FP \$UMCU/\$FILU.PF FP, PF

.Transactions file:

- User input
 - File MB

.Output file:

- Selected entry points
 - File (FLR 160, CISZ 12800) FH

.Output report(s):

- Validation report
 - SYSOUT IE

REMOVAL OF DUPLICATE ENTRY POINTS: PAN215

.Transactions file:

- Selected entry points
 - File FH

.Permanent output files:

- Sorted selected entry points
 - PRMFL : \$UMCU/\$FILU.CRIT(+1) HF
- Reduced entry points to be purged
 - PRMFL : \$UMCU/\$FILU.CRIR(+1) FR

.Sort file(s):

- File S1

PAC/IMPACT

ISEP: SELECTION OF ENTRY POINTS

ISEP: EXECUTION JCL

7

1

4

7.1.4. ISEP: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.ISEP
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: ENTRY-POINT SELECTIONS *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.ISEP *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FHO
$ SELECT $UMCU/$JCL.FRO
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.ISEP
$ FILE BM,C1S,1R
$ PAN210.
$ RUN RUFIL=$UMCS/$RUNS.PAN210,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL FP,R,R,$UMCU/$FILU.FP
$ PRMFL PF,R,R,$UMCU/$FILU.PF
$ FILE MB,C1R
$ FILE FH,C2S,100R
$ FILE IE,C3S,100L
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PAN215.
$ RUN RUFIL=$UMCS/$RUNS.PAN215
$ LIMITS 10
$ PRMFL HF,W,R,&FHO
$ PRMFL FR,W,S,&FRO
$ FILE FH,C2R
$ FILE S1,,100R
$ GOTO P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,C3R
$ SYSOUT OT,&RMTB
$ OUTPUT GBCD,MEDIA/3
$ GOTO END
$ PASCII.
$ ASCII-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,C3R
$ SYSOUT OT,&RMTA
$ OUTPUT ASCII,MEDIA/7
$ FILSYS.
$ FILSYS
CPOS $UMCU/$JCL
MF FH1,NEWNAM/FHFIL/

```

PAC/IMPACT

ISEP: SELECTION OF ENTRY POINTS

ISEP: EXECUTION JCL

7

1

4

```
MF  FH-1,NEWNAM/FH1/
MF  FH0,NEWNAM/FH-1/
MF  FHFIL,NEWNAM/FH0/
MF  FR1,NEWNAM/FRFIL/
MF  FR-1,NEWNAM/FR1/
MF  FRO,NEWNAM/FR-1/
MF  FRFIL,NEWNAM/FRO/
$  END.
$      CONVER
$      DATA      IN
***** ISEP - NORMAL END OF RUN *****
$      SYSOUT    OT,ORG
$      OUTPUT    MEDIA/03
$  ERROR.
$      ENDJOB
```

PAC/IMPACT	PAGE	259
IPEP: ENTRY-POINT PRINTOUT		7
IPEP: INTRODUCTION		2
		1

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 theabend, 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.

PAC/IMPACT
IPEP: ENTRY-POINT PRINTOUT
IPEP: DESCRIPTION OF STEPS

PAGE 260
7
2
2

7.2.2. IPEP: DESCRIPTION OF STEPS

IPEP: DESCRIPTION OF STEPS

PRINTING OUT ENTRY POINTS: PAN220

.Permanent input files:
-Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Entry points
 PRMFL : \$UMCU/\$FILU.CRIT(0) HF

.Output report:
-List of entry points
 SYSOUT IL

.Sort file(s):
 File S1

PAC/IMPACT

IPEP: ENTRY-POINT PRINTOUT

IPEP: EXECUTION JCL

7

2

3

7.2.3. IPEP: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.IPEP
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: ENTRY-POINT PRINTOUT *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FH0
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ PAN220.
$ RUN RUFIL=$UMCS/$RUNS.PAN220,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL HF,R,R,&FHI
$ FILE IL,C3S,100L
$ FILE S1,,10R
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ GOTO P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,C3R
$ SYSOUT OT,&RMTB
$ OUTPUT GBCD,MEDIA/3
$ GOTO END
$ PASCII.
$ ASCII-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,C3R
$ SYSOUT OT,&RMTA
$ OUTPUT ASCII,MEDIA/7
$ END.
$ CONVER
$ DATA IN
***** IPEP - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

PAC/IMPACT	PAGE	262
ISOS: SELECTION OF STRINGS AND OPERATORS		7
ISOS: INTRODUCTION		3
		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 conditioned ('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'.)
- . 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)').

	PAGE	263
PAC/IMPACT		7
ISOS: SELECTION OF STRINGS AND OPERATORS		3
ISOS: INTRODUCTION		1

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.

PAC/IMPACT

7

ISOS: SELECTION OF STRINGS AND OPERATORS

3

ISOS: USER INPUT

2

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)
-----

```


PAC/IMPACT

ISOS: SELECTION OF STRINGS AND OPERATORS

ISOS: USER INPUT

7

3

2

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 sepa-
!   !   !         ! rator 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 Procedu-
!   !   !         ! ral 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 Procedu-
!   !   !         ! ral code part:
!   !   !         ! .Category condition and Structure
!   !   !         ! .Source Data-Element code (Struct.)
!   !   ! ' '      ! Search in the three above mentioned
!   !   !         ! parts
-----

```

PAC/IMPACT
ISOS: SELECTION OF STRINGS AND OPERATORS
ISOS: DESCRIPTION OF STEPS

PAGE

266

7
3
3

7.3.3. ISOS: DESCRIPTION OF STEPS

ISOS: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

SELECTION OF STRINGS AND OPERATORS: PAN212

.Permanent input files:
-Error messages
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Entities in production
PRMFL : \$UMCU/\$FILU.FP \$UMCU/\$FILU.PF FP, PF

.Transaction file:
-User input
File MB

.Output file(s):
-Selected entry points
File (FLR 160, CISZ 12800) FH
-Impact analysis results
File (FLR 260, CISZ 8192) MF

.Output report(s):
-Validation report
SYSOUT IE

DELETION OF DUPLICATE ENTRY POINTS: PAN215

.Transaction file:
-Selected entry points
File FH

.Output files:
-Sorted selected entry points
File HF
-Reduced entry points to be purged
File FR

.Sort file(s):
File S1

UPDATE OF IMPACT ANALYSIS RESULTS: PAN260

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

.Permanent input file:
-Results from preceding analysis
PRMFL : \$UMCU/\$FILU.RESU(0) OF

.Permanent output file:
-Sorted impact-analysis results
PRMFL : \$UMCU/\$FILU.RESU(+1) FO

.Sort file(s):
File S1

PAC/IMPACT

ISOS: SELECTION OF STRINGS AND OPERATORS

ISOS: EXECUTION JCL

7

3

4

7.3.4. ISOS: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.ISOS
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: SELECTION OF STRINGS & OPERATORS *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.ISOS *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FHO
$ SELECT $UMCU/$JCL.FOO
$ SELECT $UMCU/$JCL.FRO
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.ISOS
$ FILE BM,C1S,1R
$ PAN212.
$ RUN RUFILE=$UMCS/$RUNS.PAN212,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL FP,R,R,$UMCU/$FILU.FP
$ PRMFL PF,R,R,$UMCU/$FILU.PF
$ FILE MB,C1R
$ FILE FH,C2S,100R
$ FILE MF,C3S,100R
$ FILE IE,C4S,100L
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PAN215.
$ RUN RUFILE=$UMCS/$RUNS.PAN215
$ LIMITS 10
$ FILE HF,W,R,&FHO
$ FILE FR,W,S,&FRO
$ FILE FH,C2R
$ FILE S1,,100R
$ PAN260.
$ RUN RUFILE=$UMCS/$RUNS.PAN260
$ LIMITS 10
$ PRMFL OF,R,R,&FOI
$ PRMFL FO,W,R,&FOO
$ FILE MF,C3R
$ FILE S1,,100R
$ GOTO P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,C4R
$ SYSOUT OT,&RMTB
$ OUTPUT GBCD,MEDIA/3
$ GOTO END
$ PASCII.
$ ASCII-PRINT 132 CH.

```

PAC/IMPACT

ISOS: SELECTION OF STRINGS AND OPERATORS

7

3

ISOS: EXECUTION JCL

4

```
$      CONVER
$      LIMITS    ,,30K
$      FILE      IN,C4R
$      SYSOUT    OT,&RM TA
$      OUTPUT    ASCII,MEDIA/7
$  FILSYS.
$      FILSYS
CPOS $UMCU/$JCL
MF    FH1,NEWNAM/FHFIL/
MF    FH-1,NEWNAM/FH1/
MF    FH0,NEWNAM/FH-1/
MF    FHFIL,NEWNAM/FH0/
MF    FO1,NEWNAM/FOFIL/
MF    FO-1,NEWNAM/FO1/
MF    FOO,NEWNAM/FO-1/
MF    FOFIL,NEWNAM/FO0/
MF    FR1,NEWNAM/FRFIL/
MF    FR-1,NEWNAM/FR1/
MF    FRO,NEWNAM/FR-1/
MF    FRFIL,NEWNAM/FR0/
$  END.
$      CONVER
$      DATA      IN
***** ISOS - NORMAL END OF RUN *****
$      SYSOUT    OT,ORG
$      OUTPUT    MEDIA/03
$  ERROR.
$      ENDJOB
```

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.

PAC/IMPACT
IMFH : MERGE FH FILES
IMFH: DESCRIPTION OF STEPS

PAGE

270

7
4
2

7.4.2. IMFH: DESCRIPTION OF STEPS

IMFH: DESCRIPTION OF STEPS

DELETION OF DUPLICATE ENTRY POINTS: PAN215

.Transaction file:
-Selected entry points

.Permanent output files:
-Sorted selected entry points

-Reduced entry points to be purged

.Sort file(s):
FILE

S1

PAC/IMPACT

IMFH : MERGE FH FILES

7

4

IMFH: EXECUTION JCL

3

7.4.3. IMFH: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.IMFH
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS : MERGE FH FILES *
$ NOTE * AND CREATION FR FILE *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FH0
$ SELECT $UMCU/$JCL.FR0
$ UTL8.
$ UTL8
$ PRMFL I1,R,R,&FHI
$ SELECT $UMCU/$JCL.FH-1
$ PRMFL I2,R,R,&FHI
$ FILE OU,C1S,100R
READ I1 I2 WRITE OU.
$ PAN215.
$ RUN RUFIL=$UMCS/$RUNS.PAN215
$ LIMITS 10
$ PRMFL HF,W,R,&FHO
$ PRMFL FR,W,S,&FRO
$ FILE FH,C1
$ FILE S1,,100R
$ FILSYS.
$ FILSYS
CPOS $UMCU/$JCL
MF FH1,NEWNAM/FHFIL/
MF FH-1,NEWNAM/FH1/
MF FH0,NEWNAM/FH-1/
MF FHFIL,NEWNAM/FH0/
MF FR1,NEWNAM/FRFIL/
MF FR-1,NEWNAM/FR1/
MF FR0,NEWNAM/FR-1/
MF FRFIL,NEWNAM/FR0/
$ END.
$ CONVER
$ DATA IN
***** IMFH - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

	PAGE	272
PAC/IMPACT		7
IANA: IMPACT SEARCH CRITERIA		5
IANA: INTRODUCTION		1

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.

	PAGE	273
PAC/IMPACT		7
IANA: IMPACT SEARCH CRITERIA		5
IANA: INTRODUCTION		1

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).

PAC/IMPACT
IANA: IMPACT SEARCH CRITERIA
IANA: DESCRIPTION OF STEPS

PAGE

274

7
5
2

7.5.2. IANA: DESCRIPTION OF STEPS

IANA: DESCRIPTION OF STEPS

RECOGNITION OF CRITERIA AFTER THE PURGE: PAN230

.Permanent input files:
-Search criteria
 PRMFL : \$UMCU/\$FILU.CRIT(0) FH
-Criteria after purge (reduced file)
 PRMFL : \$UMCU/\$FILU.CRIR(0) FR

.Output file:
-Search criteria
 File (FLR 160, CISZ 12800) HF

PRINTING OF ENTRY POINTS: PAN220

.Permanent input files:
-Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Sorted criteria
 File HF

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

.Sort file(s):
 File S1

IMPACT ANALYSIS: PAN250

.Permanent input files:
-Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
 PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-File of entities to be analyzed
 PRMFL : \$UMCU/\$FILU.FP \$UMCU/\$FILU.PF FP, PF

	PAGE	275
PAC/IMPACT		7
IANA: IMPACT SEARCH CRITERIA		5
IANA: DESCRIPTION OF STEPS		2

```

.Transaction file:
-Impacted criteria
  File                               FH

.Input-output file:
-Impacted criteria already processed
  File                               FQ, QF

.Output files:
-New impacted criteria
  File (FLR 160, CISZ 12800)         HF
-Impact analysis results
  File (FLR 260, CISZ 8192)         MF

.Return codes:
-Switch-30
  1 - Search-criteria file is empty.

```

UPDATE OF IMPACT ANALYSIS RESULTS: PAN260

```

.Transaction file:
-Impact analysis results (level)
  File                               MF

.Permanent input file:
-Results of previous analysis
  PRMFL : $UMCU/$FILU.RESU(0)       OF

.Permanent output file:
-Sorted results of impact analysis
  PRMFL : $UMCU/$FILU.RESU(+1)     FO

.Sort file(s):
  File                               S1

```

REMOVAL OF DUPLICATE ENTRY POINTS: PAN215

```

.Transaction file:
-Selected entry points
  File                               FH

.Permanent output file:
-Sorted selected entry points
  PRMFL : $UMCU/$FILU.CRIT(+1)     HF
-Reduced entry points to be purged
  PRMFL : $UMCU/$FILU.CRIR(+1)     FR

.Sort file(s):
  File                               S1

```

PAC/IMPACT

7

IANA: IMPACT SEARCH CRITERIA

5

IANA: EXECUTION JCL

3

7.5.3. IANA: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.IANA
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: IMPACT SEARCH *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FH0
$ SELECT $UMCU/$JCL.FO0
$ SELECT $UMCU/$JCL.FQ0
$ SELECT $UMCU/$JCL.FR0
$ PANFQI.
$ RUN RUFILE=$UMCS/$RUNS.PANFQI
$ LIMITS 10
$ PRMFL IN,R,R,&FQI
$ FILE FQ,Q1S,100R
$ FILE QF,Q2S,10R
$ PAN230.
$ RUN RUFILE=$UMCS/$RUNS.PAN230
$ LIMITS 10
$ PRMFL FH,R,R,&FHI
$ PRMFL FR,R,S,&FRI
$ FILE HF,C1S,10R
$ FILE S1,,10R
$ PAN220.
$ RUN RUFILE=$UMCS/$RUNS.PAN220,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE HF,C1S
$ FILE IL,C3S,100L
$ FILE S1,,10R
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PAN250.
$ RUN RUFILE=$UMCS/$RUNS.PAN250,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 100
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL FP,R,R,$UMCU/$FILU.FP
$ PRMFL PF,R,R,$UMCU/$FILU.PF
$ FILE FQ,Q1S
$ FILE QF,Q2S
$ FILE FH,C1R
$ FILE HF,C2S,100R
$ FILE MF,C3S,100R
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ IF 30,END
$ PAN260.
$ RUN RUFILE=$UMCS/$RUNS.PAN260
$ LIMITS 10
$ PRMFL OF,R,R,&FOI
$ PRMFL FO,W,R,&FOO
$ FILE MF,C3R
$ FILE S1,,100R

```

PAC/IMPACT

IANA: IMPACT SEARCH CRITERIA

7

5

IANA: EXECUTION JCL

3

```
$ PAN215.
$   RUN      RUFILE=$UMCS/$RUNS.PAN215
$   LIMITS   10
$   PRMFL    HF,W,R,&FHO
$   PRMFL    FR,W,S,&FRO
$   FILE     FH,C2R
$   FILE     S1,,100R
$ PANFQS.
$   RUN      RUFILE=$UMCS/$RUNS.PANFQS
$   LIMITS   10
$   PRMFL    OU,W,R,&FQO
$   FILE     FQ,Q1R
$   FILE     QF,Q2R
$ FILSYS.
$   FILSYS
CPOS $UMCU/$JCL
MF   FH1,NEWNAM/FHFIL/
MF   FH-1,NEWNAM/FH1/
MF   FH0,NEWNAM/FH-1/
MF   FHFIL,NEWNAM/FH0/
MF   FO1,NEWNAM/FOFIL/
MF   FO-1,NEWNAM/FO1/
MF   FO0,NEWNAM/FO-1/
MF   FOFIL,NEWNAM/FO0/
MF   FQ1,NEWNAM/FQFIL/
MF   FQ-1,NEWNAM/FQ1/
MF   FQ0,NEWNAM/FQ-1/
MF   FQFIL,NEWNAM/FQ0/
MF   FR1,NEWNAM/FRFIL/
MF   FR-1,NEWNAM/FR1/
MF   FRO,NEWNAM/FR-1/
MF   FRFIL,NEWNAM/FR0/
$ END.
$   CONVER
$   DATA    IN
***** IANA - NORMAL END OF RUN *****
$   SYSOUT   OT,ORG
$   OUTPUT   MEDIA/03
$ ERROR.
$   ENDJOB
```

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.

PAC/IMPACT	PAGE	280
IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS		7
IPIA: INTRODUCTION		6
		1

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	!

PAC/IMPACT

7

IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS

6

IPIA: USER INPUT

2

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 !
-----

```

PAC/IMPACT

7

IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS

6

IPIA: DESCRIPTION OF STEPS

3

7.6.3. IPIA: DESCRIPTION OF STEPS

IPIA: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001PRINTING OF IMPACT RESULTS: PAN270

.Permanent input files:
 -Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
 -Impact results
 PRMFL : \$UMCU/\$FILU.RESU(0) FO

.Transaction file:
 -User input
 File MB

.Output file:
 -Generated batch transactions
 File (FLR 80, CISZ 512) MV

.Output report:
 -Analysis results
 SYSOUT IF

.Sort file(s):
 File S1

PRINTING OF GENERATED TRANSACTIONS: PAN280

.Permanent input files:
 -Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Transaction file:
 -User input
 File MB
 -Generated batch transactions
 File MV

.Output files:
 -Selected batch transactions
 PRMFL : \$UMCU/\$MV.IPIA VM

.Output report
 -List of transactions by library
 SYSOUT IT

PAC/IMPACT

7

IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS

6

IPIA: EXECUTION JCL

4

7.6.4. IPIA: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.IPIA
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: RESULT PRINTOUT *
$ NOTE * * *
$ NOTE * ENTER USER INPUT *
$ NOTE * $UMCU/$MB.IPIA *
$ NOTE * * *
$ NOTE *****
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ SELECT $UMCU/$JCL.FOO
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.IPIA
$ FILE BM,C1S,1R
$ PAN270.
$ RUN RUFIL=$UMCS/$RUNS.PAN270,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL FO,R,R,&FOI
$ FILE MB,C1S
$ FILE MV,C2S,100R
$ FILE S1,,100R
$ FILE IF,D1S,100L
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PAN280.
$ RUN RUFIL=$UMCS/$RUNS.PAN280,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL VM,W,S,$UMCU/$MV.IPIA
$ FILE MV,C2R
$ FILE MB,C1R
$ FILE IT,D2S,100L
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ GOTO P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,D1R
$ FILE ",D2R
$ SYSOUT OT,&RMTB
$ OUTPUT GBCD,MEDIA/3
$ GOTO END
$ PASCII.
$ ASCII-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,D1R
$ FILE ",D2R
$ SYSOUT OT,&RMTA
$ OUTPUT ASCII,MEDIA/7

```

PAC/IMPACT

IPIA: PRINTING OF THE IMPACT ANALYSIS RESULTS

IPIA: EXECUTION JCL

7

6

4

```
$ END.  
$   CONVER  
$   DATA   IN  
***** IPIA - NORMAL END OF RUN *****  
$   SYSOUT  OT,ORG  
$   OUTPUT  MEDIA/03  
$ ERROR.  
$   ENDJOB
```

PAC/IMPACT	PAGE	286
IGRA: BREAKING DOWN OF GROUP FIELDS		7
IGRA: INTRODUCTION		7
		1

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.

	PAGE	287
PAC/IMPACT		
IGRA: BREAKING DOWN OF GROUP FIELDS		7
IGRA: INTRODUCTION		7
		1

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 analyzed. 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.

PAC/IMPACT
IGRA: BREAKING DOWN OF GROUP FIELDS
IGRA: DESCRIPTION OF STEPS

PAGE

288

7
7
2

7.7.2. IGRA: DESCRIPTION OF STEPS

IGRA: DESCRIPTION OF STEPS

RECOGNITION OF PURGED CRITERIA: PAN230

.Permanent input files:
-Search criteria file
 PRMFL : \$UMCU/\$FILU.CRIT(0) FH
-Reduced file of purged criteria
 PRMFL : \$UMCU/\$FILU.CRIR(0) FR

.Output file:
-Search criteria file
 File (FLR 160, CISZ 12800) HF

PRINTING ENTRY POINTS: PAN220

.Permanent input files:
-Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Sorted criteria
 File HF

.Output reports:
-List of accepted/eliminated criteria
 SYSOUT IL

.Sort files:
 File S1

GROUP FIELD BREAKING-DOWN: PAN255

.Permanent input files:
-Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
 PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Entities to be analyzed
 PRMFL : \$UMCB/\$BASE.FP \$UMCB/\$BASE.PF FP, PF

.Transaction file:
-Impacted criteria
 File FH

.Output file:
-Impact analysis results
 File FQ, QF

.Return codes:
 Switch-30
 1 : Fichier des critères de recherche vide

UPDATE OF IMPACT ANALYSIS RESULTS: PAN260

.Transaction file:
-Impact analysis result (by level)
 File MF

	PAGE	289
PAC/IMPACT		7
IGRA: BREAKING DOWN OF GROUP FIELDS		7
IGRA: DESCRIPTION OF STEPS		2
.Permanent input file:		
-Results of previous analysis		
PRMFL : \$UMCU/\$FILU.RESU(0)	OF	
.Permanent output file:		
-Sorted results of the impact analysis		
PRMFL : \$UMCU/\$FILU.RESU(+1)	FO	
.Sort files:		
File	S1	

PAC/IMPACT

IGRA: BREAKING DOWN OF GROUP FIELDS

IGRA: EXECUTION JCL

7

7

3

7.7.3. IGRA: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.IGRA
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: GROUP FIELDS ANALYSIS *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FH0
$ SELECT $UMCU/$JCL.FO0
$ SELECT $UMCU/$JCL.FQ0
$ SELECT $UMCU/$JCL.FR0
$ PANFQI.
$ RUN RUFILE=$UMCS/$RUNS.PANFQI
$ LIMITS 10
$ PRMFL IN,R,R,&FQI
$ FILE FQ,Q1S,100R
$ FILE QF,Q2S,10R
$ PAN230.
$ RUN RUFILE=$UMCS/$RUNS.PAN230
$ LIMITS 10
$ PRMFL FH,R,R,&FHI
$ PRMFL FR,R,S,&FRI
$ FILE HF,C1S,10R
$ FILE S1,,10R
$ PAN220.
$ RUN RUFILE=$UMCS/$RUNS.PAN220,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE HF,C1S
$ FILE IL,C3S,100L
$ FILE S1,,10R
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PAN255.
$ RUN RUFILE=$UMCS/$RUNS.PAN255,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 100
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL FP,R,R,$UMCU/$FILU.FP
$ PRMFL PF,R,R,$UMCU/$FILU.PF
$ FILE FQ,Q1S
$ FILE QF,Q2S
$ FILE FH,C1R
$ FILE HF,C2S,100R
$ FILE MF,C3S,100R
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ IF 30,END
$ PAN260.
$ RUN RUFILE=$UMCS/$RUNS.PAN260
$ LIMITS 10
$ PRMFL OF,R,R,&FOI
$ PRMFL FO,W,R,&FOO
$ FILE MF,C3R
$ FILE S1,,100R

```

PAC/IMPACT

IGRA: BREAKING DOWN OF GROUP FIELDS

IGRA: EXECUTION JCL

7

7

3

```
$ PANFQS.
$   RUN          RUFIL=$UMCS/$RUNS.PANFQS
$   LIMITS      10
$   PRMFL       OU,W,R,&FQO
$   FILE        FQ,Q1R
$   FILE        QF,Q2R
$ FILSYS.
$   FILSYS
CPOS $UMCU/$JCL
MF   FH1,NEWNAM/FHFIL/
MF   FH-1,NEWNAM/FH1/
MF   FH0,NEWNAM/FH-1/
MF   FHFIL,NEWNAM/FH0/
MF   FO1,NEWNAM/FOFIL/
MF   FO-1,NEWNAM/FO1/
MF   FO0,NEWNAM/FO-1/
MF   FOFIL,NEWNAM/FO0/
MF   FQ1,NEWNAM/FQFIL/
MF   FQ-1,NEWNAM/FQ1/
MF   FQ0,NEWNAM/FQ-1/
MF   FQFIL,NEWNAM/FQ0/
MF   FR1,NEWNAM/FRFIL/
MF   FR-1,NEWNAM/FR1/
MF   FRO,NEWNAM/FR-1/
MF   FRFIL,NEWNAM/FRO/
$ END.
$   CONVER
$   DATA      IN
***** IGRA - NORMAL END OF RUN *****
$   SYSOUT     OT,ORG
$   OUTPUT     MEDIA/03
$ ERROR.
$   ENDJOB
```

PAC/IMPACT	PAGE	292
IPFQ: FQ FILE PRINTOUT (IMPACT ANALYSIS)		7
IPFQ: INTRODUCTION		8
		1

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 search criteria.

The printout shows criteria and entry points sorted by alphabetical order within each category, and by definition library of the criteria.

The printing order for the categories are:

- . Character strings
- . Data-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.

ABNORMAL EXECUTIONS

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

PAC/IMPACT

7

IPFQ: FQ FILE PRINTOUT (IMPACT ANALYSIS)

8

IPFQ: USER INPUT

2

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 Dictio-
!   !   !      ! nary
!   !   ! '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
-----

```

PAC/IMPACT

7

IPFQ: FQ FILE PRINTOUT (IMPACT ANALYSIS)

8

IPFQ: DESCRIPTION OF STEPS

3

7.8.3. IPFQ: DESCRIPTION OF STEPS

IPFQ: DESCRIPTION OF STEPSTRANSACTION RECOGNITION: PTU001EXTRACTION OF CRITERIA: PAN240

.Permanent input files:
-Error messages

-Data file
PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Index file
PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Criteria impacted during analysis
PRMFL : \$UMCU/\$FILU.CRII(0) FQ

.Transaction file:
-Input

.Output files:
-Search criteria
File (FLR 160, CISZ 12800) FH

.Output report:
-Control report

PRINTING OF IMPACTED CRITERIA: PAN220

.Permanent input files:
-Error messages
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-Sorted entry points or criteria
File HF

.Output report:
-List of entry points or criteria
SYSOUT IL

.Sort file(s):
File S1

PAC/IMPACT

IPFQ: FQ FILE PRINTOUT (IMPACT ANALYSIS)

IPFQ: EXECUTION JCL

7

8

4

7.8.4. IPFQ: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.IPFQ
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: FQ FILE PRINTOUT *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.IPFQ *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FQO
$ GLOBAL IMP=ASCII,RMTA=($RMTA),RMTB=($RMTB)
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.IPFQ
$ FILE BM,C1S,1R
$ PAN240.
$ RUN RUFILE=$UMCS/$RUNS.PAN240,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL FQ,R,R,&FQI
$ FILE FH,C2S,100R
$ FILE MB,C1R
$ SYSOUT EI,ORG
$ SYSOUT IX,ORG
$ IF 20,ERROR
$ PAN220.
$ RUN RUFILE=$UMCS/$RUNS.PAN220,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ FILE HF,C2R
$ FILE IL,C3S,100L
$ FILE S1,,10R
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ GOTO P&IMP
$ PBCD.
$ BCD-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,C3R
$ SYSOUT OT,&RMTB
$ OUTPUT GBCD,MEDIA/3
$ GOTO END
$ PASCII.
$ ASCII-PRINT 132 CH.
$ CONVER
$ LIMITS ,,30K
$ FILE IN,C3R
$ SYSOUT OT,&RMTA
$ OUTPUT ASCII,MEDIA/7

```

	PAGE	296
PAC/IMPACT		7
INFQ: FQ FILE REINITIALIZATION (IMPACT ANALYSIS)		9
INFQ: INTRODUCTION		1

7.9. INFQ: FQ FILE REINITIALIZATION (IMPACT ANALYSIS)

7.9.1. INFQ: INTRODUCTION

INFQ: INTRODUCTION

The INFQ procedure reinitializes the FQ file, which contains search criteria that have already been impacted. Its purpose is that these criteria be ignored in future analyses.

This action should be performed before a new impact analysis when the entry points have changed or when the analysis context has changed.

However, it must not be used between two iterations of the same impact analysis.

PAC/IMPACT
INFQ: FQ FILE REINITIALIZATION (IMPACT ANALYSIS)
INFQ: DESCRIPTION OF STEPS

PAGE

297

7
9
2

7.9.2. INFQ: DESCRIPTION OF STEPS

INFO: DESCRIPTION OF STEPS

REINITIALIZATION OF THE FQ FILE: PAN200

.Output file:

-Reinitialized impactd criteria file (sequential)

PRMFL : \$UMCU/\$FILU.CRII(+1) FQ

PAC/IMPACT
 INFQ: FQ FILE REINITIALIZATION (IMPACT ANALYSIS)
 INFQ: EXECUTION JCL

7
 9
 3

7.9.3. INFQ: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.INFO
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: FQ FILE INITIALIZATION *
$ NOTE * * *
$ NOTE *****
$ SELECT $UMCU/$JCL.FQ0
$ PAN200.
$ RUN RUFIL=$UMCS/$RUNS.PAN200
$ LIMITS 10
$ PRMFL FQ,W,R,&FQ0
$ FILSYS.
$ FILSYS
CPOS $UMCU/$JCL
MF FQ1,NEWNAM/FQFIL/
MF FQ-1,NEWNAM/FQ1/
MF FQ0,NEWNAM/FQ-1/
MF FQFIL,NEWNAM/FQ0/
$ END.
$ CONVER
$ DATA IN
***** INFQ - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

	PAGE	299
PAC/IMPACT		7
INFP: FP FILE INITIALIZATION (IMPACT ANALYSIS)		10
INFP: INTRODUCTION		1

7.10. INFP: FP FILE INITIALIZATION (IMPACT ANALYSIS)

7.10.1. INFP: INTRODUCTION

INFP: INTRODUCTION

The INFP procedure initializes the FP file, which contains the selections making up the domain of the impact analysis.

For the FP file to be updated by INFP, you must re-state in the procedure's input all the lines previously introduced. As a default, the procedure initializes a blank file, i.e. containing no particular selection.

Operating principles of the FP file's input:

If an entity type is specified (whether its specific occurrences are specified or not), and you wish the analysis to take into account other types as well, you must explicitly specify those types (there again, with the `*****` code if all entities of a type are required, or specific entity codes for a narrower selection).

If an entity type is coded for all its entities --with the `*****` code-- you cannot specify a particular entity of this type.

PAC/IMPACT

INFP: FP FILE INITIALIZATION (IMPACT ANALYSIS)

INFP: USER INPUT

7

10

2

7.10.2. INFP: USER INPUT

INFP: USER INPUT

Input is optional for the INFP procedure. If no input is provided, all entities of all entity types will be searched for the impact analysis.

If all existing entities of a given entity type are specified (code = *****), particular entities specified for the same type will be refused.

```

-----
!Pos.! Len.! Value  ! Meaning
!-----+-----+-----+-----!
! 1 ! 3 !      ! Entity type
! ! !      ! Possible values are:
! ! ! 'B ' ! Database Blocks
! ! ! 'F ' ! User Entities
! ! ! 'O ' ! Screens
! ! ! 'P ' ! Programs
! ! ! 'T ' ! Texts
! ! ! 'V ' ! Volumes
! ! ! '$nn' ! User Entity Occurrence of type code!
! ! !      ! 'nn'
! ! ! '$**' ! All UEOs
! 4 ! 6 !      ! Entity code (generic selection
! ! !      ! through code *****)
! ! !      ! (This code may not exist in the
! ! !      ! Database)
-----

```

PAC/IMPACT
INFP: FP FILE INITIALIZATION (IMPACT ANALYSIS)
INFP: DESCRIPTION OF STEPS

PAGE

301

7
10
3

7.10.3. INFP: DESCRIPTION OF STEPS

INFP: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

CHECK ON TRANSACTIONS AND FP UPDATE: PAN205

.Permanent input file:
-Error messages
PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE

.Transaction file:
-User input
File MB

.Output file:
-Entities in production
PRMFL : \$UMCU/\$FILU.FP \$UMCU/\$FILU.PF FP, PF

.Output report:
-Check report
SYSOUT IP

.Sort file(s):
File S1

PAC/IMPACT

INFP: FP FILE INITIALIZATION (IMPACT ANALYSIS)

INFP: EXECUTION JCL

7

10

4

7.10.4. INFP: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.INFP
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * IMPACT ANALYSIS: FP FILE INITIALIZATION *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.INFP *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.INFP
$ FILE BM,C1S,1R
$ PAN205.
$ RUN RUFIL=$UMCS/$RUNS.PAN205,DIRFC=AA
$ DBASE PACBASE($UMCS/$SCHEMA.1STAR),
$ ETC SSPB($UMCS/$SCHEMA.SSPB)
$ LIMITS 10
$ PRMFL AA,R,S,$UMCU/$JCL.ADRUI
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL FP,L,R,$UMCU/$FILU.FP
$ PRMFL PF,L,R,$UMCU/$FILU.PF
$ FILE MB,C1R
$ FILE S1,,10R
$ SYSOUT EI,ORG
$ SYSOUT IP,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** INFP - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

VisualAge Pacbase - Operations Manual
BATCH PROCEDURES: USER'S GUIDE
VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE

PAGE 303

8

8. VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE	PAGE	304
VDWN: RESTORATION		8
VDWN: INTRODUCTION		1
		1

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'.

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VDWN: RESTORATION
 VDWN: USER INPUT

8
 1
 2

8.1.2. VDWN: USER INPUT

VDWN: USER INPUT

1. Line defining the VisualAge Pacbase library-session to be processed.

```

-----
!Pos.! Len.! Value  ! Meaning                                     ! (*) !
!-----+-----+-----+-----+-----+-----!
!  2 !  1 !  '*'  ! Line code                                     !  R  !
!-----+-----+-----+-----+-----+-----!
!  3 !  8 !      ! User code                                     !  R  !
!-----+-----+-----+-----+-----+-----!
! 11 !  8 !      ! Password                                     !  R  !
!-----+-----+-----+-----+-----+-----!
! 19 !  3 !      ! VA Pac library code                         !  R  !
!-----+-----+-----+-----+-----+-----!
! 22 !  5 !      ! Session number and status                   !  O  !
!   !   ! SPACE ! Current session                             !   !
-----

```

(*) R = Required, O = Optional

2. Extraction command line (one line per object)

```

-----
!Pos.! Len.! Value  ! Meaning                                     ! (*) !
!-----+-----+-----+-----+-----+-----!
!  2 !  2 !  'Y3' ! Line code                                     !  R  !
!-----+-----+-----+-----+-----+-----!
!  4 !  2 !      ! Object's class                             !  R  !
!   !   !  '77' ! VisualAge Smltlk. application              !   !
!-----+-----+-----+-----+-----+-----!
!  6 !  6 !      ! VA Pac identifier of the                   !  R  !
!   !   !      ! VisualAge Smalltalk object                 !   !
-----

```

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VDWN: RESTORATION
 VDWN: DESCRIPTION OF STEPS

8
 1
 3

8.1.3. VDWN: DESCRIPTION OF STEPS

VDWN: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

CHECK AND EXTRACTION PREPARATION: PVA100

.Input files:
 -Index file
 PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
 -Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
 -Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
 -User input
 File MB

.Output reports and files:
 -Check report
 SYSOUT ET
 -'*'-line check report
 SYSOUT DD
 -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. PRMFL :
 \$UMCU/\$MV.GPRT ME

-Elementary-extraction requests
 File MV

EXTRACTION: PVA110

.Input files:
 -Index file
 PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
 -Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
 -Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
 -Elementary extraction requests
 File MV

.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. PRMFL :
 \$UMCU/\$MV.VISUAL MX

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VDWN: RESTORATION
 VDWN: EXECUTION JCL

8
 1
 4

8.1.4. VDWN: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.VDWN
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * VA SMALLTALK-VA PAC BRIDGE: DOWNLOAD *
$ NOTE * * *
$ NOTE * ENTER USER INPUT IN *
$ NOTE * $UMCU/$MB.VDWN *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.VDWN
$ FILE BM,C1S,1R
$ PVA100.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PVA100
$ EXECUTE DUMP
$ LIMITS ,150K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL ME,W,S,$UMCU/$MV.GPRT
$ FILE MB,C1S
$ FILE MV,C2S,10R
$ SYSOUT DD,ORG
$ SYSOUT ET,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PVA110.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PVA110
$ EXECUTE DUMP
$ LIMITS ,150K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL MX,W,S,$UMCU/$MV.VISUAL
$ FILE MV,C2R
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** VDWN - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

	PAGE	308
VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE		8
VUP1: BACKUP - CODE CALCULATION		2
VUP1: INTRODUCTION		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.

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE	PAGE	309
VUP1: BACKUP - CODE CALCULATION		8
VUP1: INTRODUCTION		2
		1

EXECUTION CONDITIONS

None.

ABNORMAL EXECUTIONS

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

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VUP1: BACKUP - CODE CALCULATION
 VUP1: USER INPUT

8
 2
 2

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 'I*' line) with:

- . The user password
- . The Product code and the Change number, if the VisualAge Pacbase Database is under DSMS control.

```

-----
!Pos.! Len.! Value  ! Meaning                                ! (*) !
!----+-----+-----+-----+-----+-----+-----+-----!
!  2 !  2 ! 'I*'  ! Line code                                !  R  !
!----+-----+-----+-----+-----+-----+-----+-----!
!  4 !  8 !       ! User code                                !  R  !
!----+-----+-----+-----+-----+-----+-----+-----!
! 12 !  8 !       ! Password                                !  R  !
!----+-----+-----+-----+-----+-----+-----+-----!
! 20 !  3 !       ! VA Pac library code                      !  R  !
!----+-----+-----+-----+-----+-----+-----+-----!
! 23 !  5 !       ! Session number and status                !  O  !
!   !   ! SPACE ! Current session                          !    !
-----

! 58 !  9 !       ! Product + Change number if              !  O  !
!   !   !       ! database under DSMS control              !    !
-----

```

(*) R = Required, O = Optional.

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VUP1: BACKUP - CODE CALCULATION
 VUP1: USER INPUT

8
 2
 2

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 !
-----

```

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
VUP1: BACKUP - CODE CALCULATION
VUP1: DESCRIPTION OF STEPS

PAGE

312

8
2
3

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
  PRMFL : $UMCB/$BASE.AN $UMCB/$BASE.BN      AN, BN
-Data file
  PRMFL : $UMCB/$BASE.AR $UMCB/$BASE.BR      AR, BR
-Error messages
  PRMFL : $UMCB/$BASE.AE $UMCB/$BASE.XE      AE, XE
-VisualAge Smalltalk/Java file produced by workstation
  PRMFL : &FVIS                               VA

.Output reports and files:
-Check report
  SYSOUT                                     ET
- '*'-line check report
  SYSOUT                                     DD
-Extracted codes
  PRMFL : $UMCU/$FILU.VC $UMCU/$FILU.CV      VC, CV

.Sort file(s):
  File                                       S1
```

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

```
.Input files:
-Index file
  PRMFL : $UMCB/$BASE.AN $UMCB/$BASE.BN      AN, BN
-Data file
  PRMFL : $UMCB/$BASE.AR $UMCB/$BASE.BR      AR, BR
-Error message file
  PRMFL : $UMCB/$BASE.AE $UMCB/$BASE.XE      AE, XE
-VisualAge Smalltalk/Java file produced by the workstation
  PRMFL : &FVIS                               VA
-VisualAge Pacbase codes of VisualAge Smalltalk/Java
entities already saved
  PRMFL : $UMCU/$FILU.VC $UMCU/$FILU.CV      VC, CV

  PRMFL : &TCOR

.Output reports and file:
-List of new codes created
  SYSOUT                                     ET
- '*'-line check report
  SYSOUT                                     DD
-Printing of any fatal error and of the list of
duplicate entity extractions
  SYSOUT                                     ED
-List of codes assigned to new VisualAge Smalltalk/Java
entities
  PRMFL : $UMCU/$MV.PBCOD                     VN
-Useful VisualAge Smalltalk/Java transactions
  PRMFL : $UMCU/$MV.VISUTI                     VG

.Sort file(s):
  File                                       S1
```


	PAGE	313
VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE		8
VUP1: BACKUP - CODE CALCULATION		2
VUP1: DESCRIPTION OF STEPS		3

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

- .Input files:
- Index file

 - Data file

 - Error message file

 - VisualAge Smalltalk/Java file produced by the workstation

 - VisualAge Pacbase codes of VisualAge Smalltalk/Java entities already saved

 - Character-correspondence table for substitution in the code calculation
- .Output reports and file:
- List of new codes created

 - '*'-line check report

 - List of codes assigned to new VisualAge Smalltalk/Java entities

 - List of VisualAge Pacbase codes of VisualAge Smalltalk/Java entities already saved

 - List of codes assigned to the new VisualAge Smalltalk/Java entities

 - File of codes assigned to entities already stored in VisualAge Pacbase

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
VUP1: BACKUP - CODE CALCULATION
VUP1: EXECUTION JCL

8
2
4

8.2.4. VUP1: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.VUP1
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * VA SMALLTALK-VA PAC BRIDGE: UPLOAD *
$ NOTE * CODE CALCULATION *
$ NOTE * * *
$ NOTE *****
$ GLOBAL FVIS=( )
$ GLOBAL TCOR=( )
$ DEFVC.
$ FILSYS
USERID $UMCU$PWU
IGNORE ERRS
FP $UMCU/$FILU.VP
FP $UMCU/$FILU.PV
FC $UMCU/$FILU.VP,READ,LLINKS/100,2000/,MODE/RAND/
FC $UMCU/$FILU.PV,READ,LLINKS/10,200/,MODE/RAND/
$ PVA300.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PVA300
$ EXECUTE DUMP
$ LIMITS ,150K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL VA,R,S,&FVIS
$ FILE VC,V1S,100R
$ FILE S1,,10R
$ SYSOUT DD,ORG
$ SYSOUT ET,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PVA305.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PVA305
$ EXECUTE DUMP
$ LIMITS ,150K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL VA,R,S,&FVIS
$ PRMFL VC,V1S
$ PRMFL VG,W,S,$UMCU/$MV.VISUTI
$ PRMFL VN,V2S,100R
$ FILE S1,,50R
$ SYSOUT DD,ORG
$ SYSOUT ED,ORG
$ SYSOUT ET,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ PVA310.

```

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE

VUP1: BACKUP - CODE CALCULATION

8

2

VUP1: EXECUTION JCL

4

```

$      OPTION  CBL74
$      LIBRARY LA, LB
$      SELECT  $UMCS/$OBJBT.PVA310
$      EXECUTE DUMP
$      LIMITS  ,150K
$      PRMFL   1*,R/C,R,$UMCS/$SCHEMA.1STAR
$      PRMFL   LA,R/C,R,$UMCS/$FILS.OBJLIB
$      PRMFL   LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$      PRMFL   AE,Q,R,$UMCB/$BASE.AE
$      PRMFL   XE,Q,R,$UMCB/$BASE.XE
$      PRMFL   AN,Q,R,$UMCB/$BASE.AN
$      PRMFL   BN,Q,R,$UMCB/$BASE.BN
$      PRMFL   AR,Q,R,$UMCB/$BASE.AR
$      PRMFL   BR,Q,R,$UMCB/$BASE.BR
$      PRMFL   VA,R,S,&FVIS
$      PRMFL   CA,R,S,&TCOR
$      PRMFL   VC,V1R
$      PRMFL   VN,V2R
$      PRMFL   VP,W,R,$UMCU/$FILU.VP
$      PRMFL   PV,W,R,$UMCU/$FILU.PV
$      PRMFL   VV,W,S,$UMCU/$MV.PBCOD
$      FILE    S1,,50R
$      SYSOUT  DD,ORG
$      SYSOUT  ED,ORG
$      SYSOUT  ET,ORG
$      SYSOUT  EI,ORG
$      IF      20,ERROR
$  END.
$      CONVER
$      DATA   IN
$      ***** VUP1 - NORMAL END OF RUN *****
$      SYSOUT  OT,ORG
$      OUTPUT  MEDIA/03
$  ERROR.
$      ENDJOB

```

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE	PAGE	316
VUP2: GENERATION OF UPDT TRANSACTIONS		8
VUP2: INTRODUCTION		3
		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             !  O  !
!   !   !      ! database under DSMS control            !   !
!-----+-----+-----+-----+-----!

```

(*) 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

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VUP2: GENERATION OF UPDT TRANSACTIONS
 VUP2: DESCRIPTION OF STEPS

8
 3
 3

8.3.3. VUP2: DESCRIPTION OF STEPS

VUP2: DESCRIPTION OF STEPS

GENERATION OF VA PAC TRANSACTIONS FOR UPDT: PVA320

```
.Input files:
-Index file
  PRMFL : $UMCB/$BASE.AN $UMCB/$BASE.BN      AN, BN
-Data file
  PRMFL : $UMCB/$BASE.AR $UMCB/$BASE.BR      AR, BR
-Error messages
  PRMFL : $UMCB/$BASE.AE $UMCB/$BASE.XE      AE, XE
-Useful transactions produced by VisualAge Smalltalk
  (from the workstation)
  PRMFL : $UMCU/$MV.VISUTI                    VA

-Codes of new VisualAge Smalltalk/Java entities taken into
  account
  PRMFL : $UMCU/$MV.PBCOD                      VN

-Codes of VisualAge Smalltalk/Java entities already saved
  in VisualAge Pacbase
  PRMFL : $UMCU/$FILU.VC $UMCU/$FILU.CV      VC, CV

.Output reports:
-List of VisualAge Pacbase codes taken into account
  SYSOUT                                     ET
- '*'-line check report
  SYSOUT                                     DD

-List of input transactions
  SYSOUT                                     EM
-List of erroneous transactions
  SYSOUT                                     ER

.Output files:

-Transactions for UPDT that include only definitions
  File                                       MX
-Transactions for UPDT other than definitions
  File                                       MY

.Sort file(s):
  File                                       S1
```

CONCATENATION OF TRANSACTION FOR UPDT: UTL8

```
.Input files:
-File of transactions on definition files
  File                                       IN
-File of transactions other than def. files
  File                                       IN

.Output file:
-File containing the transactions for UPDT
  PRMFL : $UMCU/$MV.Vup2                    OU
```

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VUP2: GENERATION OF UPDT TRANSACTIONS
 VUP2: EXECUTION JCL

8
 3
 4

8.3.4. VUP2: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.VUP2
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * VA SMALLTALK-VA PAC BRIDGE: UPLOAD ATION*
$ NOTE * TRANSACTION GENER *
$ NOTE * *
$ NOTE *****
$ PVA320.
$ OPTION CBL74
$ LIBRARY LA, LB
$ SELECT $UMCS/$OBJBT.PVA320
$ EXECUTE DUMP
$ LIMITS ,150K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL VA,R,S,$UMCU/$MV.VISUTI
$ PRMFL VV,R,S,$UMCU/$MV.PBCOD
$ PRMFL VP,R,R,$UMCU/$FILU.VP
$ PRMFL PV,R,R,$UMCU/$FILU.PV
$ FILE MX,M1S,10L
$ FILE MY,M2S,10L
$ FILE S1,,10R
$ SYSOUT DD,ORG
$ SYSOUT ET,ORG
$ SYSOUT EM,ORG
$ SYSOUT ER,ORG
$ SYSOUT EI,ORG
$ IF 20+30,ERROR
$ UTL8.
$ UTL8
$ FILE IN,M1R
$ FILE ",M2R
$ PRMFL OU,W,S,$UMCU/$MV.VUP2
READ IN WRITE OU.
$ END.
$ CONVER
$ DATA IN
***** VUP2 - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```

	PAGE	320
VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE		
VPUR: PURGE		8
VPUR: INTRODUCTION		4
		1

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'.

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VPUR: PURGE
 VPUR: USER INPUT

8
 4
 2

8.4.2. VPUR: USER INPUT

VPUR: USER INPUT

1. VisualAge Pacbase Manager definition line:

```

-----
!Pos.! Len.! Value  ! Meaning                                ! (*) !
!-----+-----+-----+-----+-----+-----!
!  2 !  1 !  '*'  ! Line code                                !  0 !
!-----+-----+-----+-----+-----+-----!
!  3 !  8 !           ! User code                                !  0 !
!-----+-----+-----+-----+-----+-----!
! 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                                !  0 !
!-----+-----+-----+-----+-----+-----!
!  4 !  3 !           ! Code of selected library                !  0 !
-----

```

2.2. Selection of Sessions (one line for each selected session).
 If no line of this type is entered, all sessions will be selected, including the current session.

```

-----
!Pos.! Len.! Value  ! Meaning                                ! (*) !
!-----+-----+-----+-----+-----+-----!
!  2 !  2 !  'SS'  ! Line code                                !  0 !
!-----+-----+-----+-----+-----+-----!
!  4 !  5 !           ! Session code and status                !  0 !
!   !   !           ! (current session: 9999Z)                !    !
-----

```

(*) 0 = Required

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
VPUR: PURGE
VPUR: DESCRIPTION OF STEPS

PAGE

322

8
4
3

8.4.3. VPUR: DESCRIPTION OF STEPS

VPUR: DESCRIPTION OF STEPS

TRANSACTION RECOGNITION: PTU001

GENERATION OF PURGE TRANSACTIONS: PVA400

.Input files:
-Index file
 PRMFL : \$UMCB/\$BASE.AN \$UMCB/\$BASE.BN AN, BN
-Data file
 PRMFL : \$UMCB/\$BASE.AR \$UMCB/\$BASE.BR AR, BR
-Error messages
 PRMFL : \$UMCB/\$BASE.AE \$UMCB/\$BASE.XE AE, XE
-User input
 File MB

.Output reports and file:
-List of user input
 SYSOUT ET
-''-line check report
 SYSOUT DD
-Generated purge-transactions
 PRMFL : \$UMCU/\$MV.VPUR MX

.Sort file(s):
 File S1

VISUALAGE SMALLTALK/JAVA - VA PAC INTERFACE
 VPUR: PURGE
 VPUR: EXECUTION JCL

8
 4
 4

8.4.4. VPUR: EXECUTION JCL

```

$ IDENT $IDENT,$DEST.VPUR
$ NOTE *****
$ NOTE * VisualAge Pacbase *
$ NOTE * ===== *
$ NOTE * * *
$ NOTE * VA SMALLTALK-VA PAC BRIDGE: PURGE *
$ NOTE * * *
$ NOTE *****
$ PTU001.
$ OPTION CBL74
$ SELECT $UMCS/$OBJBT.PTU001
$ EXECUTE DUMP
$ LIMITS ,13K
$ PRMFL MB,R,S,$UMCU/$MB.VPUR
$ FILE BM,C1S,1R
$ PVA400.
$ OPTION CBL74
$ LIBRARY LA,LB
$ SELECT $UMCS/$OBJBT.PVA400
$ EXECUTE DUMP
$ LIMITS ,150K
$ PRMFL 1*,R/C,R,$UMCS/$SCHEMA.1STAR
$ PRMFL LA,R/C,R,$UMCS/$FILS.OBJLIB
$ PRMFL LB,R/C,S,$UMCS/$SCHEMA.CSTARPB
$ PRMFL AE,Q,R,$UMCB/$BASE.AE
$ PRMFL XE,Q,R,$UMCB/$BASE.XE
$ PRMFL AN,Q,R,$UMCB/$BASE.AN
$ PRMFL BN,Q,R,$UMCB/$BASE.BN
$ PRMFL AR,Q,R,$UMCB/$BASE.AR
$ PRMFL BR,Q,R,$UMCB/$BASE.BR
$ PRMFL MX,W,S,$UMCU/$MV.VPUR
$ FILE MB,C1R
$ FILE S1,,10R
$ SYSOUT DD,ORG
$ SYSOUT ET,ORG
$ SYSOUT EI,ORG
$ IF 20,ERROR
$ END.
$ CONVER
$ DATA IN
***** VPUR - NORMAL END OF RUN *****
$ SYSOUT OT,ORG
$ OUTPUT MEDIA/03
$ ERROR.
$ ENDJOB

```