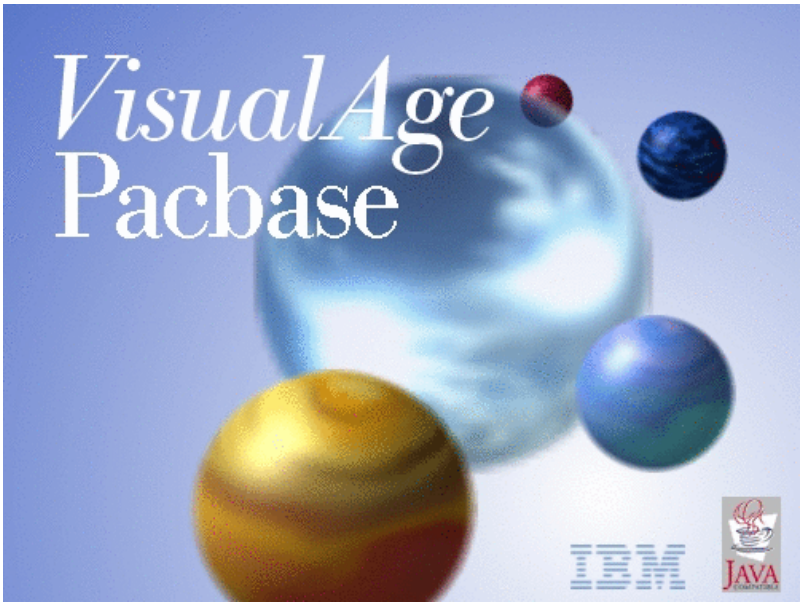
VisualAge Pacbase

Installation Guide
Z/OS CICS Server & Client Components

Version 3.5



VisualAge Pacbase

Installation Guide
Z/OS CICS Server & Client Components

Version 3.5

Note

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

You may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:

http://www.ibm.com/support/docview.wss?rs=37&uid=swg27005477

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

Fifth Edition (June 2006)

This edition applies to the following licensed programs:
- VisualAge Pacbase Version 3.5

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

IBM Paris Laboratory
1, place Jean–Baptiste Clément
93881 Noisy-le-Grand, France.

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

Contents

Notices

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504–1785, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Paris Laboratory, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

Trademarks

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

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

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

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

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

Chapter 1. Foreword

Introduction

The purpose of this manual is to guide the administrator through the installation of the VisualAge Pacbase:

- Server components
- Client components,
- Communication environment.

Once the installation is completed, it is recommended to run the set of tests provided on the installation media.

You will also find in this manual a description of the operations to be performed for the installation of correction versions.

A number of Administration actions must be carried out online in the Administrator workbench in order to make operational a VisualAge Pacbase installation or re-installation. These actions are documented in the 'AD workbench User's Guide', chapter 'Prior Administration Actions'.

VisualAge Pacbase Architecture

VisualAge Pacbase is used for the design, development and maintenance of graphical (GUI), textual (TUI) or web eBusiness applications, run in on-line or batch mode.

VisualAge Pacbase consists of:

- A server environment (TUI),
- A client environment (GUI).

These two environments communicate through an encapsulated middleware provided by IBM.

NOTE: The textual mode remains available for some functionalities.

You will find a detailed description of Server Components in chapter 'The Components' in this manual.

The Server environment

It consists of the following components:

- System elements: programs, files (online help included), and parameters.
- VA Pac Administration Database that contains user parameters and other parameters.
- one or more VA Pac Development Databases.

The Client environment

The client environment includes five separately installable components:

- Administrator & Developer workbench (also separately installable).

  Developer workbench includes the following modules (each running independently):

  – Batch,
  – Dialog,
  – eBusiness (includes three eBusiness Tools and Services Modeler),
  – Services Modeler.

  Each of these modules can be executed independently.
- eBusiness Tools:
  – Proxy Generator,
  – Location Editor,
  – Services Test Facility,
  – VisualAge Pacbase Connector,
  – VisualAge Pacbase Web Connection (Dialog Web Revamping Generator),
  – Web application models (WAM).
- VisualAge Pacbase WorkStation (Pacdesign, Pacbench)
- Web application models (WAM)
- Middleware

  The communication functions enable the Server and Client environments to communicate via the main communication protocols on the market.

Contents of Supply

The contents of the supply vary according to the terms of your order:

- Installation Guide,
- CD-Rom or cartridge, depending on the environment, to install the VA Pac server,
- Client Components CD-Rom,
- VA Pac Documentation CD-Rom.

Bibliography

- A number of administration actions must be carried out online in Aministrator workbench in order to make a VisualAge Pacbase installation or re-installation operational. These actions are documented in the 'AD workbench User's Guide', chapter 'Prior Administration Actions'.
- The procedures, used by the Administrator for the Management of Databases and versions, and utilities provided, are documented in the Administrator's procedures Guide.
- For IBM sites, the installation of a Security System, to control user codes, passwords and access authorizations, is documented in the 'Installation of Server Components' chapter, 'Additional Installations' subchapter, and also in the Online Help of Administrator workbench.

Chapter 2. Prerequisites

Prerequisites for Server Environment

Hardware and Software

Processor: IBM S/390 MVS/ESA

Installation media: 4mm DAT Drive or cartridge 3480

Monitor: CICS/ESA Version 4.1 or TS

COBOL: 'COBOL FOR OS/390 & VM 2.1.1'

Disk Space

The total amount of space needed for the files depends on the size of applications managed by the system.

The following table indicates approximately the disk space necessary (in millions of bytes) to install the servers:

Disk space for the installation	Total	VSAM	Non VSAM
Total installation	300	210	90
Total System	240	175	65
Total user files for installation tests	60	35	25

Prerequisites for Client Environment

Hardware

The hardware characteristics necessary to install VisualAge Pacbase client components are the following:

- Processor: Intel Pentium III 450 Mhz minimum or compatible processor.
- Monitor: graphic monitor (800x600) VGA or higher resolution (XGA or SVGA).
- CD-Rom drive.
- Card: adapted to the site network.
- Memory (RAM): 256 Mb (512 Mb advised).
- Software: Microsoft Windows Script Host (version 5.1 and onwards).

Disk Space

Required disk space:

- 38 Mb for Administrator & Developer workbench
- 30 Mb for eBusiness Tools
- 68 Mb for the JDK
- 15 Mb for the Middleware
- 6 Mb for WAM
- 16 Mb for the VisualAge Pacbase WorkStation (with a Methodology included)

If ″IBM SDK for Java 2″ is not installed on your workstation, its installation is automatically started.

For an operational installation of AD workbench for example, the minimum required disk space is 121 Mb (38 Mb for AD workbench, 68 Mb for the JDK and 15 for the Middleware).

Software

The VisualAge Pacbase client components require that a 32 bytes-Windows be installed on your workstation, i.e.:

- Windows 98,
- Windows/NT version 4.0 with Service Pack 3,
- Windows 2000,
- Windows XP.

See also chapter 'Installation of Client Components', subchapter 'Things to Know Before Installing'.

Communications

To enable the communication between the workstation components and the servers in a MVS/CICS, environment, the communication protocol must be: CPI-C and TCP-IP Socket.

Chapter 3. Installation of Server Components

Parameterization

SMP/E context

VA Pacbase, DSMS and Pactables use common batch and online load modules.

These load-modules are supplied in the HBVP350 root FMID.

The components specific to VA Pacbase are supplied in the JBVP351 dependent FMID.

The components specific to DSMS are supplied in the JBVP352 dependent FMID.

The components specific to Pactables are supplied in the JBVP353 dependent FMID.

To install a dependent FMID, the HBVP350 root FMID must be simultaneously or previously installed.

If the HBVP350 root FMID is already installed, only the dependent JBVP35x FMID must be installed.

On the installation cartridge, the HBVP350 root FMID and the JBVP35x dependent FMID are systematically provided.

The RECEIVE, APPLY and ACCEPT JCLs, that are supplied, are to be used for a simultaneous installation of the two FMIDs. They must be adapted if the HBVP350 root FMID is already installed.

Preparation

The preparation to the installation process consists of three stages:

- Receiving the cartridge with the SMP/E (System Modification Program/Extended) utility in dedicated PDSs.

  This stage must be performed by the system staff who are accustomed to installing IBM products with SMP/E.
- Allocation of a PDS file where all the installation and operation JCLs will be saved.
- Execution of the installation JCL from the PDS members created during the first stage.

Remark in case of a previous SMP/E installation

The following SMP/E stage implies that the SMP/E context is empty for the product: either the product has never been installed with SMP/E, or the files related to SMP/E have been reinitialized for the previous version, or you want to create a new SMP/E environment for this version.

Otherwise, if you want to re-use the SMP/E files of the previous version, you can execute, before stage 1, the SMP/E utility described in the Appendix at the end of this manual.

Stage 1

You will find details in the document 'Program Directory for VA Pac' specific to SMP/E.

If you install in a new or empty context, you will have to create the SMP/E environment and install the HBVP350 and JBVP351 FMIDs with the JCLs supplied.

This stage consists of the following steps:

- Downloading, from the cartridge, the sample JCLs necessary to the execution of steps 2, 3, and 4. Follow the instructions specified in the document titled 'Program Directory for VA Pac'.
- Setting up the SMP/E environment and the 'Target zone' and 'Distribution zone' files of the HBVP350 root FMID:
  - Defining the SMP/E cluster libraries (BVP1DCSI)
  - Initializing the SMP/E libraries (BVP2ICSI)
  - Allocating the SMP/E work files (BVP3ALLO)
  - Assigning the FMID (BVP4DEFZ)
  - Creating the DDDEF input of technical files (BVP5DDEF)
  - Defining the 'Target zone' and 'Distribution zone' files (BVP6DDEF)
  - Allocating the 'Target zone' and 'Distribution zone' files (BVP7ALLO)
- Implementing the 'Target zone' and 'Distribution zone' files of the JBVP351 dependent FMID:
  - Defining the 'Target zone' and 'Distribution zone' files (BVP8PDD)
  - Allocating the 'Target zone' and 'Distribution zone' files (BVP9PAL)
- Installing the components of the HBVP350 and JBVP351 FMIDs:
  - Execution of RECEIVE (BVPPREC)
  - Execution of APPLY (BVPPAPP)
  - Execution of ACCEPT (BVPPACC)

If you install in an environment where the HBVP350 FMID and a JBVP35x FMID are already installed, you just have to install the JBVP351 dependent FMID of VA Pacbase.

Then execute steps 3 and 4 once the necessary JCLs are downloaded as described in step 1 and modify the RECEIVE, APPLY and ACCEPT JCLs to remove the HBVP350 FMID from the impacted lines.

At the end of this first stage, all the components required for the installation are to be found in the following PDSs ; 'hlq' indicates the common prefix of the elements supplied (High-Level Qualifier):

- hlq.SBVPINST: all the 80-long files:
  - The error messages file (BVPAE)
  - The initial installation JCL (BVPTINIT)
  - A file which contains the installation and operation JCLs and procedures (BVPPTTAL)
  - The SMP/E sample JCLs.
- hlq.SBVPF2: PC and PE initialization files
- hlq.SBVPMBR8: batch load-modules (the BVPDS600 and BVPDS610 load-modules for the interface between VA Pacbase and DSMS 2.5 are provided in standard; if DSMS is to be used, you must adapt the DEXP procedure to use these new load-modules in place of PDS600 and PDS610, no longer compatible since VA Pacbase 3.0).
- hlq.SBVPMTR8: online load-modules
- hlq.SBVPF5: SA SG SN SP SR SS skeletons
- hlq.SBVPF6: SC skeleton
- hlq.SBVPF7: SF skeleton
- hlq.SBVPF8: administration model and development model update file
- hlq.SBVPSRC: source files with a length of 80
- hlq.SBVPDIC: Dictionary complementary files

Stage 2

This stage is optional but recommended. It consists in allocating a PDS file with the following characteristics:

- Lrecl=80
- Size: around 100 tracks of a 3390 disk, 30 pads directory.

Stage 3

Stage 3 consists in copying the 'hlq.SBVPINST(BVPPINIT)' JCL into the PDS mentioned in stage 2, modifying its parameters to match the constraints of the site and executing it to obtain a complete installation and operation JCL.

The BVPPINIT JCL executes the BVPMMJCL program loaded in the hlq.SBVPMBR8 PDS.

It must be completed as follows:

- Fill in '&hlq' with the value of the 'hlq' parameter used in the SMP/E first stage.
- In the '//SYSUT2 DD DSN=' field, enter the name of the file in which the complete JCL is to be saved.

  This file can be either a PDS member initially created to save all the JCLs, or a sequential file selected by the user.
- Enter the parameters (see details in the next chapter).

The BVPMMJCL program execution must be saved: it can be used for a further re-installation.

Warning

TEST and ADMIN users are automatically provided with their passwords in the Administration Database and are used in the JCLs.

The site administrator will have to delete these users after the product installation tests.

Execution JCL

```
//VAPACBAS  JOB (---),'JCL INSTALLATION',CLASS=D,MSGCLASS=A
//MM1JCL EXEC PGM=BVPMMJCL
//STEPLIB  DD DISP=SHR,DSN=&HLQ.SBVPMBR8
//         DD DISP=SHR,DSN=---.---.---    LE LIBRARY
//SYSOUT   DD SYSOUT=A
//SYSUT1   DD DSN=&HLQ.SBVPINST(BVPPTTAL),DISP=SHR
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DCB=BLKSIZE=4160
//SYSUT8   DD DUMMY,DCB=BLKSIZE=1370
//SYSUT9   DD DUMMY,DCB=BLKSIZE=1370
//*********************************************************************
//*                                                                   *
//*     CREATION OF INSTALLATION JCL THROUGH 'BVPMMJCL'               *
//*     ----------------------------------------------                *
//*                                                                   *
//*     MODIFY THE LIST OF THE SUPPLIED COMMANDS BY ASKING,           *
//*     IF NECESSARY, A SELECTION OF PARTS OF INSTALLATION JCL        *
//*     (JCL MODULES), BY GIVING THE APPROPRIATE VALUES TO THE        *
//*     INSTALLATION PARAMETERS, AND, IF NECESSARY, BY SPECIFYING     *
//*     THE LINES TO BE ADDED AT THE BEGINNING OR AT THE END OF       *
```

```
//*     EACH JCL MODULE.                                                  *
//*********************************************************************
//SYSPRM DD DUMMY
//SYSUT2   DD ---------   PDS MEMBER OR SEQUENTIAL FILE RECEIVING
//*                       THE INSTALLATION JCL (LRECL=80)
//SYSIN DD *
===PRM PRFJ=BVPJ          .JOB NAMES PREFIXES (MAX OF 5 CHARACTERS)
===PRM CCPT=<>            .JOB ACCOUNTING CODES (JOB CARDS)
===PRM CLASSJ=1           .JOB EXECUTION CLASS (JOB CARDS)
===PRM MSGCL=A            .JCL OUTPUT CLASS (MSGCLASS)
===PRM INDSV='EXP.BVP3V'   .SYSTEM VSAM FILES INDEX VA-PAC
===PRM INDSVE='EXP.BVP3VE' .SYSTEM VSAM FILES INDEX ENDEVOR
===PRM INDSN='EXP.BVP3N'   .SYSTEM NON VSAM FILES INDEX VA-PAC
===PRM INDSNE='EXP.BVP3NE' .SYSTEM NON VSAM FILES INDEX ENDEVOR
===PRM INDUV='UTI.BVP3V'   .USER VSAM FILES INDEX
===PRM INDUN='UTI.BVP3N'   .USER NON VSAM FILES INDEX VA-PAC
===PRM INDUNE='UTI.BVP3NE' .USER NON VSAM FILES INDEX ENDEVOR
===PRM BASE='BASE'         .NAME OF DATABASE DEVELOPMENT (4 CHAR)
===PRM OUT=H               .JOB SYSOUT CLASS
===PRM UWK=SYSDA           .WORK FILES UNIT
===PRM UNITSN=SYSDA        .NON VSAM SYSTEM FILES UNIT
===PRM UNITUN=SYSDA        .NON VSAM USER FILES UNIT
===PRM VOLSN=              .SYSTEM NON VSAM FILES VOL=SER=
===PRM VOLSV=              .SYSTEM VSAM FILES VOLUME
===PRM VOLUN=              .USER NON VSAM FILES VOL=SER
===PRM VOLUV=              .USER VSAM FILES VOLUME
===PRM VCAT=               .USER FILES VSAM CATALOG
===PRM SCAT=               .SYSTEM FILES VSAM CATALOG
===PRM LSK='A'             .LANGAGE OF SKELETON FILES
===PRM DSCB='BVP.DSCB'       .DSCB MODEL FILE DSNAME
===PRM BIBP='SYS1.PROCLIB'  .PROCEDURE LIBRARY
===PRM HLQ='HLQ'            .HIGH LEVEL QUALIFIER
===PRM BIBT='SYS1.SORTLIB'     .SORT LIBRARY
===PRM BCOB='SYS1.SCEERUN'     .COBOL ROUTINE LIBRARY
===PRM LDLIB='NDVR.LOADLIB'    .ENDEVOR LOADLIB (AUTHORISED LIB)
===PRM CONLIB='NDVR.CONLIB'    .ENDEVOR CONLIB
===PRM MSGLIB='NDVR.MSGLIB'    .ENDEVOR ISPMLIB
===PRM MSGSX='50'              .SUFFIX MSGS ENDEVOR ISPMLIB (00->
===PRM MSGSXH='F5F0'           .IDEM ===PRM MSGX BUT IN HEXA
===PRM CSDL='CICS410.SDFHLOAD' .DFHCSDUP STEPLIB DSN
===PRM DFHCSD='BVP.DFHCSD'     .CSD DSN
===PRM DFHEXC='BVP.DFHEXCI'    .EXCI DSN
===PRM GROUE='BVPGR'           .'GROUP' NAME FOR EXCI
===PRM GROUF='BVPGR'           .'GROUP' NAME FOR FILE
===PRM GROUP='BVPGR'           .'GROUP' NAME FOR PROGRAM
===PRM GROUT='BVPGR'           .'GROUP' NAME FOR TRANSACTION
===PRM LIST=<>                 .'LIST' NAME
===PRM TABTDF=<>               .DSN OF TABLES DESCRIPTION FILE
===PRM DSMS=NULLFILE           .DSN OF PRODUCT ELEMENTS (DC)
===PRM EXCN=BVPC               .CONNECTION FOR EXCI (4 CHAR)
===PRM EXSS=BVPS               .SESSION FOR EXCI
===PRM EXPF=EX                 .PREFIX OF VIRTUAL TERM FOR EXCI
===PRM CSOC='DBSS'            .TRANSACTION CODE "SOCKET" (4 CHAR)
===PRM CPIC='DBST'            .TRANSACTION CODE "CPIC" (4 CHAR)
```

```
===BEGMOD
./ ADD NAME=$ZMODUL
/*
//
```

JCL Installation

The BVPMMJCL module reads the JCL skeleton file and produces a complete JCL. It allows you to:

- Select portions of the JCL skeleton, which are called 'JCL modules',
- Parameterize the skeleton in order to obtain a JCL which requires a minimum of modifications to be operational,
- Add lines before and/or after the JCL modules to separate these modules.

This step can be executed as many times as necessary to generate a complete JCL.

USER INPUT

Command	Parameters	Comments
===PRM	PPPP=pppp (1)	Parameter
===SELM	jcl1 jcl2	Selected modules
===BEGMOD		Insertion of lines at the beginning of module
....1		Lines to be added before each module
....n		
===ENDMOD		Insertion of lines at the end of module
....1		Lines to be added after each module
....n		

(1) PPPP = parameter name

pppp = parameter value

Notes

- Lines ===PRM

  You can add a comment but it must be preceded by a dot and it should not exceed the column 72.

  The default values are only examples. You must therefore enter values which correspond to the choices on your site.

- Lines ===SELM

  These lines can be used to select modules.

As the standard installation provides all the modules, no particular module is selected.

- Lines ===BEGMOD

  ./ ADD NAME=$MODULE

  As a result, a line is inserted before each JCL module, in the form:

  ./ ADD NAME=<JCL-module>

RESULT: complete JCL

The file obtained in SYSUT2 contains all the installation and operation JCLs. This file must be processed via an editor to enable the installation process.

Two operations must be performed on the complete JCL:

1. Global modifications (if necessary)

Adaptations can be performed on all the JCLs.

VSAM catalogues appear as comments in the JCL obtained after the installation:

- In the DELETE/DEFINE*/
- In the JCL STEPCATs
- In the procedures' parameters

When these parameters are not required on the site, the resulting JCL can remain as it is.

When these parameters are required on the site, the affected lines should be changed into command lines. To do so, you must:

- Transform all '//*:' into '//',
- And then replace '/*:' and '*/' with blanks.

Caution: SMS

- In the installation jobs which include GenerationDataGroup allocation, you must delete the lines DD //GDGMOD from the definition IDCAMS.
- If the UNIT and VOL parameters cannot be used on the site, you can delete them in the whole JCL via an exclude command (EXCLUDE command in TSO/EDIT).

In most cases, it is recommended to perform general modifications on JCLs before the JCL splitting operation.

Caution: LSR

JCLs are standardly provided with the LSR option for the batch optimization of VSAM files access.

If the LSR option is not implemented on site, you must modify the following lines in the procedures:

```
//xxLSR DD DSN=&INDxx..file,DISP=SHR
//PACxx DD SUBSYS=(&LSR,'DDNAME=xxLSR','BUFND=10','BUFNI=10
```

by a line:

```
//PACxx DD DSN=&INDxx..file,DISP=SHR.
```

2. JCL splitting

Before each module of a standard complete JCL, there is a ./ ADD NAME=<JCL-module> line, where <JCL-module> is the code of the ===MOD line that is found (see the following table of JCL modules).

So the complete JCL can be split in as many members as JCL modules in a PDS. The complete JCL file is to be used as SYSIN for the PDS update utility: IEBUPDTE.

Note: Because of this default option, all ’./’ characters found in JCL modules containing IEBUPDTE were replaced with ’:/’.

Once the JCL is split, the replacement must be done the other way round before executing jobs which contain IEBUPDTE.

REPORT

BVPMMJCL outputs a list for each JCL module created, with the parameters taken into account and according to the variants requested.

Note:

Since the JCL skeleton parameters are formatted as $xxxx, if BVPMMJCL encounters, upon execution, a $ character which does not correspond to a defined parameter, it sends error messages such as: ’Unknown symbolic parameter’ or ’Invalid position or length’ or ’Syntax error in symbolic parameter’.

These messages do not stop the execution and should be ignored: they apply to ’$’ characters present in the flow processed by BVPMMJCL but which are NOT parameters.

List of JCLs

Table of installation JCLs

Member	Contents	Procedure
	System and Administration Database installation	
D00CSD	CICS update: components	
D01ALLOC	Allocation of parameters PDSs	
D02CPAR	Loading of common parameters	
D03DEFIN	VSAM common file definition	
D03GDG	Definition of GDG for parameter backup	
D03INI	Initialization of system files	
D04MBR	Renaming of Exit users batch load modules	
D05IPROC	Proclib allocation (optional)	
D05PROC	Loading of batch procedures in the Proclib	
D06SKEL	Loading of skeleton files	
D07AE0	Loading of error messages and documentation	INAE
D08INGU	Initialization of users code file	INGU
D08INIAD	GDG definition and initialization	
D08RSAD	Restoration of Administration Database	RSAD
D08TINQJ	Initialization of archived journal file for VA Pacbase interface	INQJ
D08XMET	Installation of administration Model	VING
D11ZXIT	Installation of Endevor exit-users	
D12SY	Loading of Endevor parameters	
D13PREP	Allocation of Endevor VSAM work file and tables	
D14EXIT	Link-edit in authorized library for Endevor	
D15MSGS	Loading of Endevor messages	
D162530	Retrieval of 2.5 files	
D99INSL	List of components' dates	INSL
	CICS complement for EXCI	
E00CSD	'Session/Connection' CSD definition	
	RACF complement	
E10RACF	Link-Edit BVPSECUR	

Member	Contents	Procedure
E11RACF	ZAP BVPSECUR	
	Installation of Development Database	
I00CSD	CICS update: components	
I01SY	Allocation of parameter PDS's	
I02SY	Loading of parameters	
I03DEF	VSAM files definition	
I03INI	GDG definition and initialization	
I04REST	Backup of tests Database	REST
I05META	Installation of development Model and configuration management	VINS
I20GDGI	Initialization of Pac/Impact GDG	

Table of operation tests JCL

Member	Contents	Procedure
	ADMINISTRATION DATABASE PROCEDURES	
JCLARAD	Archiving of the Database journal	ARAD
JCLPAGX	Extraction of Administration Database	PAGX
JCLROAD	Reorganization of Administration Database	ROAD
JCLRSAD	Restoration of Administration Database	RSAD
JCLSVAD	Backup of Administration Database	PACG
JCLUPGP	Update of PAF batch format	UPGP
	DEVELOPMENT DATABASE PROCEDURES	
JCLACTI	Activity follow-up	ACTI
JCLARCH	Archiving of the Development Database journal	ARCH
JCLCPSN	Comparison between sub-networks	PACX
JCLCSES	Compression of session number	CSES
JCLESES	Matching session numbers	ESES
JCLEMLD	Loading of error messages related to generated applications	EMLD
JCLEMUP	Update of error messages related to generated applications	EMUP
JCLEXLI	Extraction of libraries	PACX
JCLEXPJ	Extraction of transactions from journal	PACX

Member	Contents	Procedure
JCLEXTA	Extraction of an Entity	PACX
JCLEXTR	Extraction of an Entity	PACX
JCLEXUE	Extraction of a User Entity	PACX
JCLEXPU	Extraction for purge	PACX
JCLRMEN	Extraction to move an entity to a higher library, replace, rename an entity	PACX
JCLGPMC	Generation-print (MOVE CORRESPONDING)	GPMC
JCLGPRC	Generation-print (COBOL API)	GPRC
JCLGPRP	Generation-print + PPAF	GPRP
JCLGPRT	Generation-print	GPRT
JCLGPRU	Generation-print source + loading in source library	GPRU
JCLMLIB	Library management	PACS
JCLPQCA	Quality control	PQCA
JCLPQCE	Quality control extraction	PQCE
JCLPRGS	Master outline file	PRGS
JCLREOR	Reorganization of Development Database	REOR
JCLREST	Backup of Development Database	REST
JCLRESY	Complement system backup	RESY
JCLSADM	SSADM method integrity control	SADM
JCLSASN	Backup of sub-networks	PACS
JCLSASY	Complement system backup	SASY
JCLSAVE	Backup of Development Database	PACS
JCLUPDP	Update of PAF batch format	UPDP
JCLUPDT	Batch update	UPDT
JCLUXSR	Extraction of libraries	PACS
JCLXPAF	Validation of extraction master path	XPAF
JCLXPDM	Validation of master outline	XPDM
JCLYSMC	Yourdon method integrity control	YSMC
	SCM Tools Interface module	
JCLUBPM	Generation of QJ records	UBPM
JCLARPM	Archiving of SCM journal (QJ)	ARPM
JCLCHPM	Integrity Control of Events/Elements	CHPM
JCLCPPM	Comparison with extracted files	CPPM

Member	Contents	Procedure
JCLEXPM	Extraction of Database Data	EXPM
JCLGPPM	Post-generation	GPPM
JCLHIPM	Database Automatic Freeze	HIPM
JCLSIPM	Generation simulation	SIPM
	Pactables module	
JCLGETA	Generation of tables descriptions	GETA
JCLGETD	Generation of tables descriptions	GETD
JCLGETI	Initialization of tables	GETI
	PacTransfer module	
JCLTRDU	Production of DSMS transactions	TRDU
JCLTRJC	Compression of journal file	TRJC
JCLTRPF	Generation of transfer transactions	TRPF
JCLTRRP	Extraction of journal	TRRP
JCLTRUP	Update of Pac/Transfer parameters	TRUP
JCLTRED	Printing of Pac/Transfer parameters	TRED
	PAC/IMPACT MODULE	
JCLIANA	Impact analysis	IANA
JCLIGRA	Splitting up into group areas	IGRA
JCLIMFH	Merging FH creation FR	IMFH
JCLINFP	Initialization of FP	INFP
JCLINFQ	Initialization of FQ	INFQ
JCLIPEP	Print search criteria	IPEP
JCLIPFQ	Print of FQ file	IPFQ
JCLIPIA	Print of results	IPIA
JCLISEP	Selection of input	ISEP
JCLISOS	Selection of strings, operators	ISOS
	Retrieval of former releases	
RGY250	Exchange of GY transactions from 2.n vers.	GY25
RMB250	Exchange of MB transactions from 2.n vers.	MB25
RGY300	Exchange of GY transactions to 2.n vers.	GY30
RMB300	Exchange of MB transactions to 2.n vers.	MB30
RMP250	Retrieval of passwords from 2.5 vers.	UTMP
RPC250	Retrieval of PC file from 2.n version	PC25

Member	Contents	Procedure
RPE250	Retrieval of PE file from 2.n version	PE25
RPG200	Retrieval of PG file from 2.0 version	PG20
RPG250	Retrieval of PG file from 2.5 version	PG25
RPJ250	Retrieval of PJ file from 2.n version	PG25
RPP250	PEI retrieval from 2.n version onwards	PP25
RUV250	Retrieval of UV file from Pac/Transfer	UV25
	UTILITIES	
JCLPURGE	Purge of LB PA WS work files	
JCLSTAT	Generation statistics file on base	STAT
	'P' lines extraction with 'UNS' operator	UTU1
	'P' lines update with 'UNS' operator	UTU2
	Endevor	
	Intra-Endevor integrity validation	CEND
	Inter-environments integrity validation	CIND
	Recognition of Endevor parameters	GPND
	Retrieval of the 2.n archived journal	JJND
	Retrieval of archived journal transactions	JRND
	Generation of VA Pac transaction in QJ	MEND
	Infopac elements creation (retrieval)	RIND
	VA Pac elements creation (retrieval)	RPND
	Retrieval of 2.n Types file	RPTY
	Retrieval of user entities	RP25
	Existing data retrieval	RRND
	Loading of 'TY' types VSAM file	TYND
	VA Pacbase elements import into Endevor	UPND

JCL Parameters

Syntax

===PRM PPPP=pppp .Comments

- Parameter values which contain special characters must be entered with simple quotes.
- Comments on ===PRM lines must not exceed the column 72

  They must be preceded by a dot ('.').

Note: When the default or '<>' values are filled in, the parameter must be set.

Table of parameters

Parameter	Meaning	Default
	JOB lines:	
PRFJ	Jobname prefix (maximum: 5 characters)	BVPJ
CCPT	Job accounting code	<>
CLASSJ	Job execution class	1
MSGCL	JCL output class	A
INDSV	VA Pac VSAM system (maximum: 24 characters)	'EXP.BVP3V'
INDSVE	Endevor VSAM system (maximum: 24 characters)	'EXP.BVP3VE'
INDSN	VA Pac Non-VSAM system (SAM,PDS) (maximum: 24 characters)	'EXP.BVP3N'
INDSNE	Endevor Non-VSAM system (SAM,PDS) (maximum: 24 characters)	'EXP.BVP3NE'
INDUV	VSAM user (maximum: 24 characters)	'UTI.BVP3V'
INDUN	Non VSAM user (SAM) (Maximum: 24 characters)	'UTI.BVP3N'
BASE	Development Database code (it is automatically loaded in the development transaction code field)	BASE
OUT	SYSOUT print class	H
UWK	UNIT of work files used	SYSDA
UNITSN	UNIT of non-VSAM system files	SYSDA
UNITUN	UNIT of NON VSAM user files	SYSDA
VOLSN	VOL=SER of NON-VSAM files	
VOLSV	VOL=SER of VSAM system files	
VOLUN	VOL=SER of NON-VSAM user file	
VOLUV	VOL=SER of VSAM user files	
VCAT	VSAM catalog of the Development Database (user files)	
SCAT	VSAM system catalog (System files)	
LSK	Skeleton language (1)	A
DSCB	DSNAME of DSCB model file used for files at generation	'BVP.DSCB'

Parameter	Meaning	Default
BIBP	DSNAME of the procedure library	'SYS1.PROCLIB'
HLQ	Common prefix of elements delivered with SMP/E (maximum 30 characters)	'HLQ'
BIBT	DSNAME of sort library (SORTLIB)	'SYS1.SORTLIB'
BCOB	DSNAME of COBOL routine library	'SYS1.SCEERUN'
	Update of CICS CDS	
LDLIB	Endevor load-lib (authorized Lib.)	'NDVR.LOADLIB'
CONLIB	Endevor CONLIB	'NDVR.CONLIB'
MSGLIB	Endevor MSGLIB	'NDVR.MSGLIB'
MSGSX	Endevor messages Suffix	'50'
MSGSXH	Endevor messages Suffix, in hexa.	'F5F0'
CSDL	DFHCSDUP STEPLIB DSN	'CICS410.SDFHLOAD'
DFHCSD	CSD name for update	'BVP.DFHCSD'
DFHEXC	EXCI modules PDS (4) (5)	'BVP.DFHEXCI'
	CSD input group	
GROUE	EXCI (4)	BVPGR
GROUP	Programs	BVPGR
GROUF	Files	BVPGR
GROUT	Transactions	BVPGR
LIST	List containing CICS groups	<>
	Pactables	
TABTDF	Table description file DSN	TD300
	DSMS module (2)	
DSMS	Development Database-element file name (2)	NULLFILE
	EXCI (4)	
EXCN	EXCI connection (4)	BVPC
EXSS	EXCI session (4)	BVPS
EXPF	Prefix of EXCI virtual terminal (4)	EX
	Transactions	
CSOC	SOCKET transaction code (3)	DBSS
CPIC	CPIC transaction code (3)	DBST

(1) A = English, F = French

(2) These files DSNs should be replaced with those installed on site only if the Tables or DSMS modules are installed or if you want to change the default name.

(3) These transaction codes are related to the use of Aministrator and Developer workbench. One of them is used to access the communication monitor for a TCP-IP Sockets protocol, the other is used to access the communication monitor for a CPI-C protocol. This code must be indicated in the installation parameters of the client environment.

(4) These parameters must be replaced with those installed on site if the SCM module is used.

(5) If SCM is used, the asterisk (which indicates a comment) must be removed from the line which contains the DFHEXC parameter, in the STEPLIB of the PAC step, in the GPRC, GPRP, GPRT and GPRU procedures.

Separators of JCL Modules

Lines before and after JCL modules

```
===BEGMOD
....1    )
.....    ) Lines to be inserted before each JCL module
....n    )
===ENDMOD
....1    )
.....    ) Lines to be inserted after each JCL module
....n    )
```

Lines may be inserted as input to BVPMMJCL if the default option is not appropriate (see Subchapter 'Installation default options' above).

The purpose of these lines is to execute the separation of the JCL file created by the BVPMMLCL utility into as many members as there are JCL modules.

This utility adds ....1 to ....n lines in front of each JCL module and ....1 to ....n lines to the end of each JCL module.

System Installation

CICS CSD Update

System components

D00CSD module: '$prfj.D0' job

BVP: SMP/E root, prefix of the Administration Database, batch and on-line load-modules and procedures files.

Files	Comments
BVPAE	Error messages - on-line documentation
BVPGU	User codes / keywords
BVPGN	Administration Database, index file
BVPGR	Administration Database, data file
BVPGJ	Administration Database, transaction file
BVPGY	Administration Database, random data
BVPLB	Job follow-up file
BVPPA	PAF working file
BVPQJ	SCM working file
BVPTR	A&D workbench working file
BVPWS	A&D workbench working file

Programs

BVPQnnn, BVPPnnn, BVPRnnn, BVPSnnn and a few specific programs prefixed with BVP.

Note: The 'Dynamic Backout' option (JNLSYNCWrite and RECOVery) is required for GN, GR, GJ, and GY files.

Addition of programs under CICS

It is required to add, in 'DFHRPL', the DD line of the VA Pac online programs library.

For more details on the declaration of programs, refer to the 'Efficiency enhancement' subchapter in the Appendix of this manual.

Uppercase and lowercase management

The VA Pacbase manages the input character as follows:

- All the codes entered in lowercase are automatically transformed into uppercase,
- All entity names, as well as texts, remain in lowercase. Implicit keywords drawn from entity names are changed into uppercase words.

In order to inhibit the lowercase to uppercase switch, the letter 'X' should be entered in the ACTION CODE field.

This case management is provided by the BVPUTCR program.

To work in uppercase letters only, you just have to replace the BVPUTCR program with the BVPUTCX program.

Execution JCL

```
//$PRFJ.D0 JOB ($CCPT),'SYSTEM DFHCSDUP',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - CICSCSD                              *
//*           CICS: CSD BATCH UPDATE SYSTEM                           *
//*********************************************************************
//DFHCSDUP EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=$CSDL,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DFHCSD   DD DSN=$DFHCSD,DISP=SHR
//SYSIN    DD *
*********************************************************************
*                    COMMON FILES                                   *
*********************************************************************
DEFINE FILE(BVPAE) GROUP($GROUF)
DESCRIPTION(HELP FILE)
       DSNAME($INDSV..BVPAE)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(NO)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPPA) GROUP($GROUF)
DESCRIPTION(PAF FILE)
       DSNAME($INDSV..BVPPA)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPGN) GROUP($GROUF)
DESCRIPTION(INDEX ADMINISTRATION DATABASE)
       DSNAME($INDSV..BVPGN)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPGR) GROUP($GROUF)
```

```
DESCRIPTION(DATA ADMINISTRATION DATABASE)
       DSNAME($INDSV..BVPGR)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPGY) GROUP($GROUF)
DESCRIPTION(EXTEND ADMINISTRATION DATABASE)
       DSNAME($INDSV..BVPGY)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPGJ) GROUP($GROUF)
DESCRIPTION(JOURNAL ADMINISTRATION DATABASE)
       DSNAME($INDSV..BVPGJ)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPGU) GROUP($GROUF)
DESCRIPTION(USER ADMINISTRATION DATABASE)
       DSNAME($INDSV..BVPGU)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(V)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPLB) GROUP($GROUF)
DESCRIPTION(FILE LB)
       DSNAME($INDSV..BVPLB)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(NO) BROWSE(YES) DELETE(NO) READ(YES) UPDATE(NO)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPQJ) GROUP($GROUF)
DESCRIPTION(FILE QJ)
       DSNAME($INDSV..BVPQJ)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPTR) GROUP($GROUF)
DESCRIPTION(FILE TR)
```

```
       DSNAME($INDSV..BVPTR)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE(BVPWS) GROUP($GROUF)
DESCRIPTION(FILE WS)
       DSNAME($INDSV..BVPWS)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
*****************************************************************
DEFINE PROGRAM(BVPCICSE) GROUP($GROUP)
                         EXECKEY(CICS)
DEFINE PROGRAM(BVPCMPUF) GROUP($GROUP)
DEFINE PROGRAM(BVPFAJO) GROUP($GROUP)
DEFINE PROGRAM(BVPFANM) GROUP($GROUP)
DEFINE PROGRAM(BVPFBAG) GROUP($GROUP)
DEFINE PROGRAM(BVPFBIB) GROUP($GROUP)
DEFINE PROGRAM(BVPFCHA) GROUP($GROUP)
DEFINE PROGRAM(BVPFCHK) GROUP($GROUP)
DEFINE PROGRAM(BVPFCTL) GROUP($GROUP)
DEFINE PROGRAM(BVPFDBD) GROUP($GROUP)
DEFINE PROGRAM(BVPFECR) GROUP($GROUP)
DEFINE PROGRAM(BVPFENU) GROUP($GROUP)
DEFINE PROGRAM(BVPFFOG) GROUP($GROUP)
DEFINE PROGRAM(BVPFIAP) GROUP($GROUP)
DEFINE PROGRAM(BVPFMCL) GROUP($GROUP)
DEFINE PROGRAM(BVPFPGM) GROUP($GROUP)
DEFINE PROGRAM(BVPFRUB) GROUP($GROUP)
DEFINE PROGRAM(BVPFRWS) GROUP($GROUP)
DEFINE PROGRAM(BVPFSDO) GROUP($GROUP)
DEFINE PROGRAM(BVPFTER) GROUP($GROUP)
DEFINE PROGRAM(BVPFTRA) GROUP($GROUP)
DEFINE PROGRAM(BVPFTXT) GROUP($GROUP)
DEFINE PROGRAM(BVPFVER) GROUP($GROUP)
DEFINE PROGRAM(BVPF000) GROUP($GROUP)
DEFINE PROGRAM(BVPQA00) GROUP($GROUP)
DEFINE PROGRAM(BVPQB00) GROUP($GROUP)
DEFINE PROGRAM(BVPQC00) GROUP($GROUP)
DEFINE PROGRAM(BVPQC01) GROUP($GROUP)
DEFINE PROGRAM(BVPQC50) GROUP($GROUP)
DEFINE PROGRAM(BVPQD00) GROUP($GROUP)
DEFINE PROGRAM(BVPQE00) GROUP($GROUP)
DEFINE PROGRAM(BVPQF00) GROUP($GROUP)
DEFINE PROGRAM(BVPQF10) GROUP($GROUP)
DEFINE PROGRAM(BVPQG00) GROUP($GROUP)
DEFINE PROGRAM(BVPQH00) GROUP($GROUP)
DEFINE PROGRAM(BVPQH01) GROUP($GROUP)
DEFINE PROGRAM(BVPQH20) GROUP($GROUP)
```

```
DEFINE PROGRAM(BVPQH30) GROUP($GROUP)
DEFINE PROGRAM(BVPQI00) GROUP($GROUP)
DEFINE PROGRAM(BVPQI01) GROUP($GROUP)
DEFINE PROGRAM(BVPQI02) GROUP($GROUP)
DEFINE PROGRAM(BVPQI03) GROUP($GROUP)
DEFINE PROGRAM(BVPQI04) GROUP($GROUP)
DEFINE PROGRAM(BVPQI05) GROUP($GROUP)
DEFINE PROGRAM(BVPQI20) GROUP($GROUP)
DEFINE PROGRAM(BVPQI21) GROUP($GROUP)
DEFINE PROGRAM(BVPQI50) GROUP($GROUP)
DEFINE PROGRAM(BVPQK10) GROUP($GROUP)
DEFINE PROGRAM(BVPQK20) GROUP($GROUP)
DEFINE PROGRAM(BVPQK30) GROUP($GROUP)
DEFINE PROGRAM(BVPQL10) GROUP($GROUP)
DEFINE PROGRAM(BVPQL20) GROUP($GROUP)
DEFINE PROGRAM(BVPQL21) GROUP($GROUP)
DEFINE PROGRAM(BVPQL30) GROUP($GROUP)
DEFINE PROGRAM(BVPQL40) GROUP($GROUP)
DEFINE PROGRAM(BVPQL41) GROUP($GROUP)
DEFINE PROGRAM(BVPQL45) GROUP($GROUP)
DEFINE PROGRAM(BVPQL46) GROUP($GROUP)
DEFINE PROGRAM(BVPQM00) GROUP($GROUP)
DEFINE PROGRAM(BVPQP00) GROUP($GROUP)
DEFINE PROGRAM(BVPQP01) GROUP($GROUP)
DEFINE PROGRAM(BVPQP02) GROUP($GROUP)
DEFINE PROGRAM(BVPQP03) GROUP($GROUP)
DEFINE PROGRAM(BVPQP04) GROUP($GROUP)
DEFINE PROGRAM(BVPQP05) GROUP($GROUP)
DEFINE PROGRAM(BVPQP06) GROUP($GROUP)
DEFINE PROGRAM(BVPQP07) GROUP($GROUP)
DEFINE PROGRAM(BVPQP08) GROUP($GROUP)
DEFINE PROGRAM(BVPQP50) GROUP($GROUP)
DEFINE PROGRAM(BVPQQR0) GROUP($GROUP)
DEFINE PROGRAM(BVPQR00) GROUP($GROUP)
DEFINE PROGRAM(BVPQS02) GROUP($GROUP)
DEFINE PROGRAM(BVPQS03) GROUP($GROUP)
DEFINE PROGRAM(BVPQS04) GROUP($GROUP)
DEFINE PROGRAM(BVPQS05) GROUP($GROUP)
DEFINE PROGRAM(BVPQS06) GROUP($GROUP)
DEFINE PROGRAM(BVPQS07) GROUP($GROUP)
DEFINE PROGRAM(BVPQT00) GROUP($GROUP)
DEFINE PROGRAM(BVPQT10) GROUP($GROUP)
DEFINE PROGRAM(BVPQT20) GROUP($GROUP)
DEFINE PROGRAM(BVPQT50) GROUP($GROUP)
DEFINE PROGRAM(BVPQU10) GROUP($GROUP)
DEFINE PROGRAM(BVPQU20) GROUP($GROUP)
DEFINE PROGRAM(BVPQV10) GROUP($GROUP)
DEFINE PROGRAM(BVPQV20) GROUP($GROUP)
DEFINE PROGRAM(BVPQV30) GROUP($GROUP)
DEFINE PROGRAM(BVPQX00) GROUP($GROUP)
DEFINE PROGRAM(BVPQX01) GROUP($GROUP)
DEFINE PROGRAM(BVPQY01) GROUP($GROUP)
DEFINE PROGRAM(BVPQY02) GROUP($GROUP)
DEFINE PROGRAM(BVPQY03) GROUP($GROUP)
DEFINE PROGRAM(BVPQY04) GROUP($GROUP)
```

```
DEFINE PROGRAM(BVPQY05) GROUP($GROUP)
DEFINE PROGRAM(BVPQY10) GROUP($GROUP)
DEFINE PROGRAM(BVPQY11) GROUP($GROUP)
DEFINE PROGRAM(BVPQY20) GROUP($GROUP)
DEFINE PROGRAM(BVPQY30) GROUP($GROUP)
DEFINE PROGRAM(BVPQY33) GROUP($GROUP)
DEFINE PROGRAM(BVPQZ00) GROUP($GROUP)
DEFINE PROGRAM(BVPQ000) GROUP($GROUP)
DEFINE PROGRAM(BVPQ100) GROUP($GROUP)
DEFINE PROGRAM(BVPQ200) GROUP($GROUP)
DEFINE PROGRAM(BVPQ210) GROUP($GROUP)
DEFINE PROGRAM(BVPQ300) GROUP($GROUP)
DEFINE PROGRAM(BVPQ400) GROUP($GROUP)
DEFINE PROGRAM(BVPQ500) GROUP($GROUP)
DEFINE PROGRAM(BVPQ600) GROUP($GROUP)
DEFINE PROGRAM(BVPQ700) GROUP($GROUP)
DEFINE PROGRAM(BVPQ800) GROUP($GROUP)
DEFINE PROGRAM(BVPQ900) GROUP($GROUP)
DEFINE PROGRAM(BVPRACF) GROUP($GROUP)
DEFINE PROGRAM(BVPRADM) GROUP($GROUP)
DEFINE PROGRAM(BVPRBAS) GROUP($GROUP)
DEFINE PROGRAM(BVPRCE00) GROUP($GROUP)
DEFINE PROGRAM(BVPRCS00) GROUP($GROUP)
DEFINE PROGRAM(BVPRC100) GROUP($GROUP)
DEFINE PROGRAM(BVPRDOC) GROUP($GROUP)
DEFINE PROGRAM(BVPRDO00) GROUP($GROUP)
DEFINE PROGRAM(BVPRENU) GROUP($GROUP)
DEFINE PROGRAM(BVPRIT00) GROUP($GROUP)
DEFINE PROGRAM(BVPRKEY) GROUP($GROUP)
DEFINE PROGRAM(BVPRLGF) GROUP($GROUP)
DEFINE PROGRAM(BVPRLGS) GROUP($GROUP)
DEFINE PROGRAM(BVPRLIC) GROUP($GROUP)
DEFINE PROGRAM(BVPRMC00) GROUP($GROUP)
DEFINE PROGRAM(BVPRMS00) GROUP($GROUP)
DEFINE PROGRAM(BVPROP00) GROUP($GROUP)
DEFINE PROGRAM(BVPRPAR) GROUP($GROUP)
DEFINE PROGRAM(BVPRPRC) GROUP($GROUP)
DEFINE PROGRAM(BVPRPVP) GROUP($GROUP)
DEFINE PROGRAM(BVPRSEC) GROUP($GROUP)
DEFINE PROGRAM(BVPRS12) GROUP($GROUP)
DEFINE PROGRAM(BVPRTRF) GROUP($GROUP)
DEFINE PROGRAM(BVPRUSE) GROUP($GROUP)
DEFINE PROGRAM(BVPRVL00) GROUP($GROUP)
DEFINE PROGRAM(BVPRXX00) GROUP($GROUP)
DEFINE PROGRAM(BVPRWB00) GROUP($GROUP)
DEFINE PROGRAM(BVPR000) GROUP($GROUP)
DEFINE PROGRAM(BVPR005) GROUP($GROUP)
DEFINE PROGRAM(BVPR100) GROUP($GROUP)
DEFINE PROGRAM(BVPR200) GROUP($GROUP)
DEFINE PROGRAM(BVPR400) GROUP($GROUP)
DEFINE PROGRAM(BVPR500) GROUP($GROUP)
DEFINE PROGRAM(BVPR600) GROUP($GROUP)
DEFINE PROGRAM(BVPR980) GROUP($GROUP)
                        RESIDENT(YES)
DEFINE PROGRAM(BVPR990) GROUP($GROUP)
```

```
                        RESIDENT(YES)
DEFINE PROGRAM(BVPSA00) GROUP($GROUP)
DEFINE PROGRAM(BVPSB00) GROUP($GROUP)
DEFINE PROGRAM(BVPSB02) GROUP($GROUP)
DEFINE PROGRAM(BVPSB10) GROUP($GROUP)
DEFINE PROGRAM(BVPSB30) GROUP($GROUP)
DEFINE PROGRAM(BVPSCPI) GROUP($GROUP)
DEFINE PROGRAM(BVPSC00) GROUP($GROUP)
DEFINE PROGRAM(BVPSC10) GROUP($GROUP)
DEFINE PROGRAM(BVPSD00) GROUP($GROUP)
DEFINE PROGRAM(BVPSD05) GROUP($GROUP)
DEFINE PROGRAM(BVPSD07) GROUP($GROUP)
DEFINE PROGRAM(BVPSD10) GROUP($GROUP)
DEFINE PROGRAM(BVPSD15) GROUP($GROUP)
DEFINE PROGRAM(BVPSD16) GROUP($GROUP)
DEFINE PROGRAM(BVPSD17) GROUP($GROUP)
DEFINE PROGRAM(BVPSD20) GROUP($GROUP)
DEFINE PROGRAM(BVPSD25) GROUP($GROUP)
DEFINE PROGRAM(BVPSECT) GROUP($GROUP)
DEFINE PROGRAM(BVPSERR) GROUP($GROUP)
DEFINE PROGRAM(BVPSE00) GROUP($GROUP)
DEFINE PROGRAM(BVPSE10) GROUP($GROUP)
DEFINE PROGRAM(BVPSE20) GROUP($GROUP)
DEFINE PROGRAM(BVPSE30) GROUP($GROUP)
DEFINE PROGRAM(BVPSFA0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFB0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFC0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFD0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFE0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFF0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFH0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFIN) GROUP($GROUP)
DEFINE PROGRAM(BVPSFI0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFJ0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFK0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFL0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFO0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFP0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFQ0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFR0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFS0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFT0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFV0) GROUP($GROUP)
DEFINE PROGRAM(BVPSFX0) GROUP($GROUP)
DEFINE PROGRAM(BVPSF00) GROUP($GROUP)
DEFINE PROGRAM(BVPSF05) GROUP($GROUP)
DEFINE PROGRAM(BVPSF07) GROUP($GROUP)
DEFINE PROGRAM(BVPSF10) GROUP($GROUP)
DEFINE PROGRAM(BVPSF20) GROUP($GROUP)
DEFINE PROGRAM(BVPSH00) GROUP($GROUP)
DEFINE PROGRAM(BVPSH02) GROUP($GROUP)
DEFINE PROGRAM(BVPSH04) GROUP($GROUP)
DEFINE PROGRAM(BVPSH10) GROUP($GROUP)
DEFINE PROGRAM(BVPSH20) GROUP($GROUP)
DEFINE PROGRAM(BVPSH30) GROUP($GROUP)
```

```
DEFINE PROGRAM(BVPSH40) GROUP($GROUP)
DEFINE PROGRAM(BVPSH42) GROUP($GROUP)
DEFINE PROGRAM(BVPSH44) GROUP($GROUP)
DEFINE PROGRAM(BVPSI00) GROUP($GROUP)
DEFINE PROGRAM(BVPSI10) GROUP($GROUP)
DEFINE PROGRAM(BVPSI20) GROUP($GROUP)
DEFINE PROGRAM(BVPSI30) GROUP($GROUP)
DEFINE PROGRAM(BVPSJ00) GROUP($GROUP)
DEFINE PROGRAM(BVPSK00) GROUP($GROUP)
DEFINE PROGRAM(BVPSL00) GROUP($GROUP)
DEFINE PROGRAM(BVPSO00) GROUP($GROUP)
DEFINE PROGRAM(BVPSO02) GROUP($GROUP)
DEFINE PROGRAM(BVPSO03) GROUP($GROUP)
DEFINE PROGRAM(BVPSO04) GROUP($GROUP)
DEFINE PROGRAM(BVPSO05) GROUP($GROUP)
DEFINE PROGRAM(BVPSO10) GROUP($GROUP)
DEFINE PROGRAM(BVPSO12) GROUP($GROUP)
DEFINE PROGRAM(BVPSO37) GROUP($GROUP)
DEFINE PROGRAM(BVPSO38) GROUP($GROUP)
DEFINE PROGRAM(BVPSO80) GROUP($GROUP)
DEFINE PROGRAM(BVPSP00) GROUP($GROUP)
DEFINE PROGRAM(BVPSP10) GROUP($GROUP)
DEFINE PROGRAM(BVPSP20) GROUP($GROUP)
DEFINE PROGRAM(BVPSP30) GROUP($GROUP)
DEFINE PROGRAM(BVPSP40) GROUP($GROUP)
DEFINE PROGRAM(BVPSP50) GROUP($GROUP)
DEFINE PROGRAM(BVPSP60) GROUP($GROUP)
DEFINE PROGRAM(BVPSP70) GROUP($GROUP)
DEFINE PROGRAM(BVPSP80) GROUP($GROUP)
DEFINE PROGRAM(BVPSQ00) GROUP($GROUP)
DEFINE PROGRAM(BVPSQ05) GROUP($GROUP)
DEFINE PROGRAM(BVPSQ07) GROUP($GROUP)
DEFINE PROGRAM(BVPSR00) GROUP($GROUP)
DEFINE PROGRAM(BVPSR10) GROUP($GROUP)
DEFINE PROGRAM(BVPSR20) GROUP($GROUP)
DEFINE PROGRAM(BVPSR30) GROUP($GROUP)
DEFINE PROGRAM(BVPSR40) GROUP($GROUP)
DEFINE PROGRAM(BVPSR50) GROUP($GROUP)
DEFINE PROGRAM(BVPSSOC) GROUP($GROUP)
DEFINE PROGRAM(BVPSS00) GROUP($GROUP)
DEFINE PROGRAM(BVPSS1B) GROUP($GROUP)
DEFINE PROGRAM(BVPSS1C) GROUP($GROUP)
DEFINE PROGRAM(BVPSS10) GROUP($GROUP)
DEFINE PROGRAM(BVPSS20) GROUP($GROUP)
DEFINE PROGRAM(BVPSS30) GROUP($GROUP)
DEFINE PROGRAM(BVPSS40) GROUP($GROUP)
DEFINE PROGRAM(BVPSS50) GROUP($GROUP)
DEFINE PROGRAM(BVPST00) GROUP($GROUP)
DEFINE PROGRAM(BVPST10) GROUP($GROUP)
DEFINE PROGRAM(BVPST20) GROUP($GROUP)
DEFINE PROGRAM(BVPST30) GROUP($GROUP)
DEFINE PROGRAM(BVPSV00) GROUP($GROUP)
DEFINE PROGRAM(BVPSV10) GROUP($GROUP)
DEFINE PROGRAM(BVPSV20) GROUP($GROUP)
DEFINE PROGRAM(BVPSV30) GROUP($GROUP)
```

```
DEFINE PROGRAM(BVPSX00) GROUP($GROUP)
DEFINE PROGRAM(BVPSX10) GROUP($GROUP)
DEFINE PROGRAM(BVPSY00) GROUP($GROUP)
DEFINE PROGRAM(BVPSY05) GROUP($GROUP)
DEFINE PROGRAM(BVPSY07) GROUP($GROUP)
DEFINE PROGRAM(BVPSY10) GROUP($GROUP)
DEFINE PROGRAM(BVPSY20) GROUP($GROUP)
DEFINE PROGRAM(BVPSY30) GROUP($GROUP)
DEFINE PROGRAM(BVPSY40) GROUP($GROUP)
DEFINE PROGRAM(BVPSY50) GROUP($GROUP)
DEFINE PROGRAM(BVPTPST) GROUP($GROUP)
DEFINE PROGRAM(BVPTPWS) GROUP($GROUP)
DEFINE PROGRAM(BVPUCTR) GROUP($GROUP)
                        RESIDENT(YES)
DEFINE PROGRAM(BVPUCTX) GROUP($GROUP)
DEFINE PROGRAM(BVPUFAA) GROUP($GROUP)
DEFINE PROGRAM(BVPUFB1) GROUP($GROUP)
DEFINE PROGRAM(BVPUFB2) GROUP($GROUP)
DEFINE PROGRAM(BVPUFE2) GROUP($GROUP)
*****************************************************************
DEFINE TRANSACTION($CSOC) GROUP($GROUT)
DESCRIPTION(SOCKET TRANSACTION CODE) PROGRAM(BVPSSOC)
DEFINE TRANSACTION($CPIC) GROUP($GROUT)
DESCRIPTION(CPIC TRANSACTION CODE) PROGRAM(BVPSCPI)
*****************************************************************
ADD GROUP($GROUF) LIST($LIST)
ADD GROUP($GROUP) LIST($LIST)
ADD GROUP($GROUT) LIST($LIST)
//
```

Allocation and Loading of System Parameters

D01ALLOC module: '$prfj.D1' job

Allocation of file: $INDSN..BVPSY

Step	Program	Comments
STEP1	IDCAMS	DELETE of files
STEP2	IEFBR14	allocation of files

Execution JCL

```
//$PRFJ.D1   JOB ($CCPT),'PAC D01ALLOC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*******************************************************************
//* VISUALAGE PACBASE                                               *
//*                                                                 *
//*               INSTALLATION - D01ALLOC                           *
//* INITIAL ALLOCATION OF THE PARAMETER PDS AND ADDITIONAL FILES     *
//*     .STEP1 : LISTCAT                                            *
//*     .STEP2 : ALLOCATION                                         *
//*******************************************************************
//*
```

```
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  LISTC ENT($INDSN..BVPSY)
/*
//STEP2  EXEC PGM=IEFBR14,COND=(0,EQ,STEP1)
//SY       DD DSN=$INDSN..BVPSY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(50,,5))
//
```

D02CPAR module: '$prfj.D2C' job

Loading of $INDSN..BVPSY file

Step	Program	Comments
STEP1	IEBUPDTE	Loading of PDS members

Caution

Replace all :/ with ./ before submitting the job.

Execution JCL

```
===FRM TYPE=DATA
//$PRFJ.D2C  JOB ($CCPT),'PAC D02CPAR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D02CPAR                              *
//*    LOADS PDS OF COMMON PARAMETERS                                 *
//*     .STEP1 : LOADING COMMON PARAMETERS                            *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDSN..BVPSY,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
:/     ADD NAME=SRTREO1
 SORT FIELDS=(1,25,A,48,4,A,32,7,A,39,1,D,54,7,A,26,1,D),FORMAT=BI
:/     ADD NAME=SRTREO2
 SORT FIELDS=(1,60,A),FORMAT=BI
:/     ADD NAME=SRTPC25
 SORT FIELDS=(42,8,A),FORMAT=BI
:/     ADD NAME=DFBVPAE
 DELETE ($INDSV..BVPAE) CLUSTER
 SET LASTCC = 0
```

```
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPAE)           -
                  SHR (2,3) RUS   KEYS (12,0)    -
                  VOL ($VOLSV)                   -
                  CYL (45)                       -
                  RECSZ (80 80) )                -
        INDEX   ( NAME ($INDSV..BVPAE.I)         -
                  CISZ (1024) )                  -
        DATA    ( NAME ($INDSV..BVPAE.D)         -
                  FSPC (2,1)                     -
        CISZ (4096) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPGN
 DELETE ($INDSV..BVPGN) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPGN)           -
                  SHR (2) RUS   KEYS (49,0)      -
                  VOL ($VOLSV)                   -
                  CYL (4 1)                      -
                  RECSZ (59,59) )                -
        INDEX   ( NAME ($INDSV..BVPGN.I)         -
                  CISZ (4096) )                  -
        DATA    ( NAME ($INDSV..BVPGN.D)         -
                  FSPC (10,5)                    -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPGR
 DELETE ($INDSV..BVPGR) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPGR)           -
                  SHR (2) RUS   NUMBERED         -
                  VOL ($VOLSV)                   -
                  CYL (4 1)                      -
                  RECSZ (144,144)  )             -
       DATA    ( NAME ($INDSV..BVPGR.D)          -
       CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPGY
 DELETE ($INDSV..BVPGY) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPGY)           -
                  SHR (2) RUS   NUMBERED         -
                  VOL ($VOLSV)                   -
                  CYL (4 1)                      -
                  RECSZ (1018,1018)  )           -
       DATA    ( NAME ($INDSV..BVPGY.D)          -
       CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPGJ
 DELETE ($INDSV..BVPGJ) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPGJ)           -
                  SHR (2) RUS   NUMBERED         -
                  VOL ($VOLSV)                   -
                  CYL (1 1)                      -
```

```
                  RECSZ (170,170)  )              -
        DATA    ( NAME ($INDSV..BVPGJ.D)          -
        CISZ (4096) )        /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPGU
 DELETE ($INDSV..BVPGU) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPGU)          -
                  SHR (2,3) RUS   KEYS (8,0)      -
                  VOL ($VOLSV)                    -
                  CYL (4 1)                       -
                  RECSZ (80,80) )                 -
        INDEX   ( NAME ($INDSV..BVPGU.I)          -
                  CISZ (1024) )                   -
        DATA    ( NAME ($INDSV..BVPGU.D)          -
                  FSPC (2,1)                      -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPQJ
 DELETE ($INDSV..BVPQJ) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPQJ)            -
                  SHR (3 3)   RUS   NUMBERED      -
                  VOL ($VOLSV)                    -
                  CYL (4 1)                       -
                  RECSZ (1105,1105) )               -
        DATA    ( NAME ($INDSV..BVPQJ.D)          -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPSC
 DELETE ($INDSV..BVPSC) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPSC)          -
                  SHR (2,3) RUS   KEY (4 0)       -
                  VOL ($VOLSV)                    -
                  RECSZ (3204,3204) )             -
        INDEX   ( NAME ($INDSV..BVPSC.I)          -
                  CISZ (256) )                    -
        DATA    ( NAME ($INDSV..BVPSC.D)          -
                  FSPC (10,5) SPEED  TRK (5)      -
        CISZ (3584) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPSA
 DELETE ($INDSV..BVPSA) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPSA)          -
                  SHR (2,3) RUS   KEY (5 0)       -
                  VOL ($VOLSV)                    -
                  RECSZ (4605,4605) )             -
        INDEX   ( NAME ($INDSV..BVPSA.I)          -
                  CISZ (256) )                    -
        DATA    ( NAME ($INDSV..BVPSA.D)          -
                  FSPC (10,5) SPEED  CYL (4 1)    -
        CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPSG
 DELETE ($INDSV..BVPSG) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPSG)          -
                  SHR (2,3) RUS   KEY (5 0)       -
                  VOL ($VOLSV)                    -
                  RECSZ (4605,4605) )             -
        INDEX   ( NAME ($INDSV..BVPSG.I)          -
```

```
                CISZ (256) )                    -
        DATA    ( NAME ($INDSV..BVPSG.D)        -
                FSPC (10,5) SPEED  CYL (4 1)    -
        CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPSN
 DELETE ($INDSV..BVPSN) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPSN)        -
                SHR (2,3) RUS   KEY (5 0)       -
                VOL ($VOLSV)                    -
                RECSZ (4605,4605) )             -
        INDEX   ( NAME ($INDSV..BVPSN.I)        -
                CISZ (256) )                    -
        DATA    ( NAME ($INDSV..BVPSN.D)        -
                FSPC (10,5) SPEED  CYL (5 1)    -
        CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPSR
 DELETE ($INDSV..BVPSR) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPSR)        -
                SHR (2,3) RUS    KEY (5 0)      -
                VOL ($VOLSV)                    -
                RECSZ (4605,4605) )             -
        INDEX   ( NAME ($INDSV..BVPSR.I)        -
                CISZ (256) )                    -
        DATA    ( NAME ($INDSV..BVPSR.D)        -
                FSPC (10,5) SPEED  TRK (5)      -
        CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPSP
 DELETE ($INDSV..BVPSP) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPSP)        -
                SHR (2,3) RUS   KEY (5 0)       -
                VOL ($VOLSV)                    -
                RECSZ (4605,4605) )             -
        INDEX   ( NAME ($INDSV..BVPSP.I)        -
                CISZ (256) )                    -
        DATA    ( NAME ($INDSV..BVPSP.D)        -
                FSPC (10,5) SPEED  TRK (3 1)    -
        CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPGS
 DELETE ($INDSV..BVPGS) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..BVPGS)          -
                SHR (2) RUS   KEYS (25,0)       -
                VOL ($VOLSV)                    -
                CYL (1 1)                       -
                RECSZ (214,214) )               -
        INDEX   ( NAME ($INDSV..BVPGS.I)        -
                CISZ (1024) )                   -
        DATA    ( NAME ($INDSV..BVPGS.D)        -
                FSPC (10,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPSS
 DELETE ($INDSV..BVPSS) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPSS)        -
                SHR (2,3) RUS   KEY (5 0)       -
                VOL ($VOLSV)                    -
                RECSZ (4605,4605) )             -
```

```
        INDEX   ( NAME ($INDSV..BVPSS.I)          -
                  CISZ (256) )                    -
        DATA    ( NAME ($INDSV..BVPSS.D)          -
                  FSPC (10,5) SPEED  CYL (5 1)    -
        CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPGK
 DELETE ($INDSV..BVPGK) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPGK)          -
                  SHR (4) RUS     KEY (29 0)      -
                  VOL ($VOLSV)                    -
                  CYL (4 1)                       -
                  RECSZ (80,80) )                 -
        INDEX   ( NAME ($INDSV..BVPGK.I)          -
                  CISZ (1024) )                   -
        DATA    ( NAME ($INDSV..BVPGK.D)          -
                  FSPC (10,5)                     -
        CISZ (4096) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFTABTDF
 DELETE ($TABTDF) CLUSTER
 DEFINE CLUSTER ( NAME ($TABTDF)                  -
                  SHR (2 3) RUS   KEYS (21,0)     -
                  VOL ($VOLUV)                    -
                  CYL (1 1)                       -
                  RECSZ (240,240) )               -
        INDEX   ( NAME ($TABTDF..I)               -
                  CISZ (1024) )                   -
        DATA    ( NAME ($TABTDF..D)               -
                  FSPC (10,5)                     -
        CISZ (2048) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFDSMSDC
 LISTCAT ENTRIES ($DSMS) CLUSTER
 IF LASTCC NE 0 THEN DO
 DELETE ($DSMS) CLUSTER
 DEFINE CLUSTER ( NAME ($DSMS)                    -
                  SHR (2,3) REUSE  KEYS (31,2)    -
                  VOL ($VOLUV)                    -
                  CYL (1 1)                       -
                  RECSZ (50,168) )                -
        INDEX   ( NAME ($DSMS..I)                 -
                  CISZ (1024) )                   -
        DATA    ( NAME ($DSMS..D)                 -
                  FSPC (10,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
 END
:/     ADD NAME=DFBVPLB
 DELETE ($INDSV..BVPLB) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPLB)            -
                  SHR (4)     RUS   KEYS (23,0)   -
                  VOL ($VOLSV)                    -
                  CYL (2 1)                       -
                  RECSZ (117,117) )               -
        INDEX   ( NAME ($INDSV..BVPLB.I)          -
                  CISZ (4096) )                   -
```

```
        DATA    ( NAME ($INDSV..BVPLB.D)          -
                  FSPC (10,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFBVPPA
 DELETE ($INDSV..BVPPA) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..BVPPA)            -
                  SHR (2 3) RUS  KEYS (37 2)      -
                  VOL ($VOLSV)                    -
                  CYL (5 1)                       -
                  RECSZ (200 1161) )              -
        INDEX   ( NAME ($INDSV..BVPPA.I)          -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDSV..BVPPA.D)          -
                  FSPC (10 5) SPEED               -
        CISZ (16384) )        /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPTR
 DELETE ($INDSV..BVPTR) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..BVPTR)            -
                  SHR (2 3)  RUS  NUMBERED        -
                  VOL ($VOLSV)                    -
                  CYL (1 1)                       -
                  RECSZ (4000 4000) )             -
        DATA    ( NAME ($INDSV..BVPTR.D)          -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPWS
 DELETE ($INDSV..BVPWS) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..BVPWS)            -
                  SHR (2 3) RUS  KEYS (37 0)      -
                  VOL ($VOLSV)                    -
                  CYL (150 5)                     -
                  RECSZ (1537 1537) )             -
        INDEX   ( NAME ($INDSV..BVPWS.I)          -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDSV..BVPWS.D)          -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($SCAT) */
:/     ADD NAME=DFSYSPAF
 DELETE (&USES.SYSPAF.&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME (&USES.SYSPAF.&USER)     -
                  SHR (2 3)  RUS  KEYS(12 0)      -
                  VOL ($VOLUV)                    -
                  CYL (5 1)                       -
                  RECSZ (170 1031) )              -
        INDEX   ( NAME (&USES.SYSPAF.&USER.I)   -
                  CISZ (512) )                    -
        DATA    ( NAME (&USES.SYSPAF.&USER.D)   -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFSYSEXT
 DELETE (&USES.SYSEXT.&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
```

```
 DEFINE CLUSTER ( NAME (&USES.SYSEXT.&USER)     -
                  SHR (2 3)  RUS  KEYS(43 0)      -
                  VOL ($VOLUV)                    -
                  CYL (5 1)                       -
                  RECSZ (254 254) )               -
        INDEX   ( NAME (&USES.SYSEXT.&USER.I)   -
                  CISZ (512) )                    -
        DATA    ( NAME (&USES.SYSEXT.&USER.D)   -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFSYIANA
 DELETE ($INDUV..SYIANA.&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..SYIANA.&USER)     -
                  SHR (2 3)  RUS  KEYS(94 1)      -
                  VOL ($VOLUV)                    -
                  CYL (5 1)                       -
                  RECSZ (100 100) )               -
        INDEX   ( NAME ($INDUV..SYIANA.&USER.I)   -
                  CISZ (512) )                    -
        DATA    ( NAME ($INDUV..SYIANA.&USER.D)   -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFWKREOR
 DELETE ($INDUV..WK&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..WK&USER)          -
                  SHR (2 3)  RUS  KEYS(8 0)       -
                  VOL ($VOLUV)                    -
                  CYL (1 1)                       -
                  RECSZ (162,162) )               -
        INDEX   ( NAME ($INDUV..WK&USER.I)        -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDUV..WK&USER.D)        -
                  FSPC (10 5) SPEED               -
        CISZ (16384) )        /*: CATALOG ($VCAT) */
:/     ADD NAME=DFWYREOR
 DELETE ($INDUV..WY&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..WY&USER)          -
                  SHR (2 3)  RUS  KEYS(8 0)       -
                  VOL ($VOLUV)                    -
                  CYL (2 1)                       -
                  RECSZ (1028,1028) )             -
        INDEX   ( NAME ($INDUV..WY&USER.I)        -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDUV..WY&USER.D)        -
                  FSPC (10 5) SPEED               -
        CISZ (16384) )        /*: CATALOG ($VCAT) */
:/     ADD NAME=DFWKROAD
 DELETE ($INDSV..WKROAD) CLUSTER
 SET LASTCC = 0
```

```
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..WKROAD)           -
                  SHR (2 3)  RUS  KEYS(8 0)       -
                  VOL ($VOLSV)                    -
                  CYL (1 1)                       -
                  RECSZ (162,162) )               -
        INDEX   ( NAME ($INDSV..WKROAD.I)         -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDSV..WKROAD.D)         -
                  FSPC (10 5) SPEED               -
        CISZ (16384) )        /*: CATALOG ($VCAT) */
:/     ADD NAME=DFWYROAD
 DELETE ($INDSV..WYROAD) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..WYROAD)           -
                  SHR (2 3)  RUS  KEYS(8 0)       -
                  VOL ($VOLSV)                    -
                  CYL (2 1)                       -
                  RECSZ (1028,1028) )             -
        INDEX   ( NAME ($INDSV..WYROAD.I)         -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDSV..WYROAD.D)         -
                  FSPC (10 5) SPEED               -
        CISZ (16384) )        /*: CATALOG ($VCAT) */
:/     ADD NAME=DFSYTRPF
 DELETE ($INDUV..SYTRPF.&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..SYTRPF.&USER)     -
                  SHR (2 3)  RUS  KEYS(28 0)      -
                  VOL ($VOLUV)                    -
                  CYL (5 1)                       -
                  RECSZ (180 180) )               -
        INDEX   ( NAME ($INDUV..SYTRPF.&USER.I)   -
                  CISZ (512) )                    -
        DATA    ( NAME ($INDUV..SYTRPF.&USER.D)   -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DFSYTRDU
 DELETE ($INDUV..SYTRDU.&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..SYTRDU.&USER)     -
                  SHR (2 3)  RUS  KEYS(28 0)      -
                  VOL ($VOLUV)                    -
                  CYL (5 1)                       -
                  RECSZ (180 180) )               -
        INDEX   ( NAME ($INDUV..SYTRDU.&USER.I)   -
                  CISZ (512) )                    -
        DATA    ( NAME ($INDUV..SYTRDU.&USER.D)   -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DLSYSPAF
 DELETE (&USES.SYSPAF.&USER) CLUSTER
```

```
:/     ADD NAME=DLSYSEXT
 DELETE (&USES.SYSEXT.&USER) CLUSTER
:/     ADD NAME=DLSYIANA
 DELETE ($INDUV..SYIANA.&USER) CLUSTER
:/     ADD NAME=DLSYTRDU
 DELETE ($INDUV..SYTRDU.&USER) CLUSTER
:/     ADD NAME=DLWKREOR
 DELETE ($INDUV..WK&USER) CLUSTER
:/     ADD NAME=DLWYREOR
 DELETE ($INDUV..WY&USER) CLUSTER
:/     ADD NAME=DLWKROAD
 DELETE ($INDSV..WKROAD) CLUSTER
:/     ADD NAME=DLWYROAD
 DELETE ($INDSV..WYROAD) CLUSTER
:/     ADD NAME=DLPQCE
 DELETE ($INDUN..PQCE.&USER)
:/     ADD NAME=MAXKEY
99999999999999999999999999999999999999999999999999999999999999999999999999
:/     ADD NAME=KEY01
00000001
:/     ADD NAME=REPRO999
 REPRO INFILE (MAXKEY)   OUTFILE (SYSPAF)
:/     ADD NAME=REPRO
 REPRO INFILE (IN1)    OUTFILE (OU1)
:/     ADD NAME=VERIFGN
 VERIFY FILE (PACGGN)
:/     ADD NAME=VERIFGR
 VERIFY FILE (PACGGR)
:/     ADD NAME=VERIFGY
 VERIFY FILE (PACGGY)
:/     ADD NAME=VERIFGJ
 VERIFY FILE (PACGGJ)
:/     ADD NAME=VERIFGU
 VERIFY FILE (PACGGU)
:/     ADD NAME=VERIFAN
 VERIFY FILE (PAC7AN)
:/     ADD NAME=VERIFAR
 VERIFY FILE (PAC7AR)
:/     ADD NAME=VERIFAY
 VERIFY FILE (PAC7AY)
:/     ADD NAME=VERIFAJ
 VERIFY FILE (PAC7AJ)
:/     ADD NAME=VERIFAE
 VERIFY FILE (PAC7AE)
:/     ADD NAME=VERIFLB
 VERIFY FILE (PAC7LB)
:/     ADD NAME=VERIFQJ
 VERIFY FILE (PAC7QJ)
:/     ADD NAME=VERIFSA
 VERIFY FILE (PAC7SA)
:/     ADD NAME=VERIFSG
 VERIFY FILE (PAC7SG)
:/     ADD NAME=VERIFSN
 VERIFY FILE (PAC7SN)
:/     ADD NAME=VERIFSC
```

```
 VERIFY FILE (PAC7SC)
:/     ADD NAME=VERIFSR
 VERIFY FILE (PAC7SR)
:/     ADD NAME=VERIFSP
 VERIFY FILE (PAC7SP)
:/     ADD NAME=VERIFSS
 VERIFY FILE (PAC7SS)
:/     ADD NAME=VERIFTD
 VERIFY FILE (PAC7TD)
:/     ADD NAME=VERIFEM
 VERIFY FILE (PAC7EM)
:/     ADD NAME=LIBVPGJ
 LISTCAT ENTRIES ($INDSV..BVPGJ) CLUSTER
PP
//
```

Initialization

D03DEFIN module: '$prfj.D3' job

Allocation of system and Administration Database files

Step	Program	Comments
STEP1	IDCAMS	Delete/Define

Execution JCL

```
//$PRFJ.D3  JOB ($CCPT),'PAC D03DEFIN',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D03DEFIN                             *
//*    DEFINITION OF COMMON VSAM FILES                                *
//*     STEP1 : DELETE/DEFINE                                         *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN DD DSN=$INDSN..BVPSY(DFBVPSA),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPSC),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPSG),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPSN),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPSR),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPSP),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPSS),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPPA),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPGS),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPGN),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPGR),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPGY),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPGJ),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPGK),DISP=SHR
```

```
//      DD DSN=$INDSN..BVPSY(DFBVPLB),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPTR),DISP=SHR
//      DD DSN=$INDSN..BVPSY(DFBVPWS),DISP=SHR
//*
//
```

D03GDG module: '$prfj.D3G' job

Allocation of model DSCB - GDG definition

Step	Program	Comments
STEP1	IDCAMS	DELETE of model DSCB files
STEP2	IEFBR14	Allocation of model DSCB

Execution JCL

```
//$PRFJ.D3G JOB ($CCPT),'PAC D03GDG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D03GDG                               *
//*   JOB TO RUN ONLY THE FIRST TIME THE PRODUCT IS INSTALLED         *
//*    . BUILDING OF DSCB MODEL                                       *
//*********************************************************************
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($DSCB)
//*
//STEP2  EXEC PGM=IEFBR14
//DSCB     DD DISP=(,CATLG),SPACE=(TRK,0),

//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         DSN=$DSCB
//*
//
```

D03INI module: '$prfj.D3I' job

Initialization of system and Administration Database files

Step	Program	Comments
STEP1	IDCAMS	Initialization of PA work file max. key (PAF user on-line request)
STEP2	IDCAMS	Initialization of GS extraction schema file max. key (PAF extension)

Step	Program	Comments
STEP3	IDCAMS	Initialization of WS work file max. key (Workstation)
STEP4	IDCAMS	Initialization of LB file max. key (jobs follow-up)
STEP5	IDCAMS	Initialization of the generator work file max. key
PINTR	IDCAMS	Initialization of TR work file max. key (Workstation)

Execution JCL

```
//$PRFJ.D3I JOB ($CCPT),'PAC D03INI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D03INI                               *
//*   INITIALIZATION: JOB TO RUN ONLY                                 *
//*   THE FIRST TIME THE PRODUCT IS INSTALLED                         *
//*    . INITIALIZATION OF P.A.F. WORK FILES PA                       *
//*    . INITIALIZATION OF PDM EXTENSION-USER FILE "GS"               *
//*    . INITIALIZATION OF WORK FILE STATION WS                       *
//*    . INITIALIZATION OF JOB REPORT FILE LB                         *
//*    . INITIALIZATION OF GENERATION RIGHTS FILE GK                  *
//*    . INITIALIZATION OF AD WORKBENCH FILE TR                       *
//*********************************************************************
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//PAO      DD DSN=$INDSV..BVPPA,DISP=SHR
//PAI      DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (PAI)   OUTFILE (PAO)
//*
//STEP2  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//GSO      DD DSN=$INDSV..BVPGS,DISP=SHR
//GSI      DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (GSI)   OUTFILE (GSO)
//*
//STEP3  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//WSO      DD DSN=$INDSV..BVPWS,DISP=SHR
//WSI      DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (WSI)   OUTFILE (WSO)
//*
//STEP4  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
```

```
//SYSPRINT DD SYSOUT=$OUT
//LBO      DD DSN=$INDSV..BVPLB,DISP=SHR
//LBI      DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (LBI)   OUTFILE (LBO)
//*
//STEP5  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//GKO      DD DSN=$INDSV..BVPGK,DISP=SHR
//GKI      DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (GKI)   OUTFILE (GKO)
//*
//PTINTR EXEC PGM=BVPTINTR
//*----------------------
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//*:        DD DSN=$SCAT,DISP=SHR
//STEPLIB   DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//PAC7TR    DD DSN=$INDSV..BVPTR,DISP=SHR
//SYSOUT    DD SYSOUT=$OUT
//SYSUDUMP  DD SYSOUT=$OUT
//
```

Renaming the Exit Users Batch Programs

D04MBR module: '$prfj.D4B' job

Rename Exit users batch programs

Step	Program	Comments
STEP1	IEBCOPY	RENAME Exit users load modules
STEP2	IEBCOPY	COPY Exit users load modules

Execution JCL

```
//$PRFJ.D4B JOB ($CCPT),'PAC D04MBR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D04MBR                               *
//*       - RENAME EXIT USERS BATCH LOAD MODULES -                    *
//*********************************************************************
//STEP1    EXEC PGM=IEBCOPY,REGION=0M
//SYSPRINT DD SYSOUT=$OUT
//I1       DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//O1       DD DSN=&&TEMP,DISP=(NEW,PASS),
//         UNIT=SYSDA,SPACE=(CYL,(20,10,100)),
//         DCB=(TRTCH=NOCOMP,RECFM=U,BLKSIZE=6144)
//SYSIN    DD *
   C INDD=I1,OUTDD=O1
```

```
     S M=((BVPRBASB,BVPRBAS))
     S M=((BVPRC10B,BVPRC100))
     S M=((BVPRCE0B,BVPRCE00))
     S M=((BVPRCS0B,BVPRCS00))
     S M=((BVPRDOCB,BVPRDOC))
     S M=((BVPRDO0B,BVPRDO00))
     S M=((BVPRIT0B,BVPRIT00))
     S M=((BVPRKEYB,BVPRKEY))
     S M=((BVPRLGFB,BVPRLGF))
     S M=((BVPRLGSB,BVPRLGS))
     S M=((BVPRLICB,BVPRLIC))
     S M=((BVPRMC0B,BVPRMC00))
     S M=((BVPRMS0B,BVPRMS00))
     S M=((BVPROP0B,BVPROP00))
     S M=((BVPRPARB,BVPRPAR))
     S M=((BVPRPRCB,BVPRPRC))
     S M=((BVPRPVPB,BVPRPVP))
     S M=((BVPRSECB,BVPRSEC))
     S M=((BVPRTRFB,BVPRTRF))
     S M=((BVPRXX0B,BVPRXX00))
     S M=((BVPRUSEB,BVPRUSE))
     S M=((BVPRVL0B,BVPRVL00))
     S M=((BVPRWB0B,BVPRWB00))
/*
//*
//STEP2    EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//I1       DD DSN=&&TEMP,DISP=(OLD,DELETE)
//O1       DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//SYSIN    DD *
   C   I=((I1,R)),O=O1
/*
```

Loading of Procedures

It is recommended that all operation procedures be cataloged in one procedures library:

- Either in a reserved PROCLIB: in this case, execute the allocation job first, and then the loading job.
- Or in an existing PROCLIB: in this case, execute the loading job straight away.

D05IPROC module: '$prfj.D5I' job

Allocation of a reserved library (optional)

Step	Program	Comments
STEP1	IEFBR14	Allocation of procedures library

Execution JCL

```
//$PRFJ.D5I  JOB ($CCPT),'PAC D05IPROC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D05IPROC                             *
//*                                                                   *
//*                WARNING! OPTIONAL JOB                              *
//*                ====================                               *
//*                                                                   *
//*     INITIAL ALLOCATION OF A SPECIAL "PROCLIB" FOR THE PRODUCT     *
//*     .STEP1 : LISTCAT                                              *
//*     .STEP2 : ALLOCATION                                           *
//*                                                                   *
//*********************************************************************
//*
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  LISTC ENT($BIBP)
/*
//STEP2  EXEC PGM=IEFBR14,COND=(0,EQ,STEP1)
//LIB      DD DSN=$BIBP,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(200,20,10))
```

D05PROC module: '$prfj.D5P' job

Loading of procedures

This job includes an IEBUPDTE step, which creates one member for each procedure.

Caution:

Replace all :/ with ./ before submitting the job.

Each member is coded 'BVPNNNN', where NNNN is the standard name of the procedure.

Procedures are detailed in the 'Administrator's Procedures' manual or in the 'Developer's Procedures' manual, except for the Database retrieval procedures, which are described in this manual.

Execution JCL

```
//$PRFJ.D5P JOB ($CCPT),'PAC D05PROC',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
```

```
//*                                                                   *
//*               INSTALLATION - D05PROC                              *
//*                                                                   *
//*    CATALOGING OF BATCH PROCEDURES                                 *
//*                                                                   *
//*  ->NOTE:                                                          *
//*    REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                     *
//*                                                                   *
//*********************************************************************
//UPD    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$BIBP,DISP=SHR
//SYSIN    DD DATA,DLM='F+'
:/ ADD NAME=BVPACTI
:/ ADD NAME=BVPARAD
:/ ADD NAME=BVPARCH
:/ ADD NAME=BVPARPM
:/ ADD NAME=BVPCEND
:/ ADD NAME=BVPCHPM
:/ ADD NAME=BVPCIND
:/ ADD NAME=BVPCINN
:/ ADD NAME=BVPCPPM
:/ ADD NAME=BVPCSES
:/ ADD NAME=BVPEMLD
:/ ADD NAME=BVPEMUP
:/ ADD NAME=BVPESES
:/ ADD NAME=BVPEXPM
:/ ADD NAME=BVPGETA
:/ ADD NAME=BVPGETD
:/ ADD NAME=BVPGETI
:/ ADD NAME=BVPGPND
:/ ADD NAME=BVPGPMC
:/ ADD NAME=BVPGPPM
:/ ADD NAME=BVPGPRN
:/ ADD NAME=BVPGPRP
:/ ADD NAME=BVPGPRC
:/ ADD NAME=BVPGPRT
:/ ADD NAME=BVPGPRU
:/ ADD NAME=BVPGPRX
:/ ADD NAME=BVPGPRY
:/ ADD NAME=BVPGY25
:/ ADD NAME=BVPGY30
:/ ADD NAME=BVPHIPM
:/ ADD NAME=BVPIANA
:/ ADD NAME=BVPIGRA
:/ ADD NAME=BVPIMFH
:/ ADD NAME=BVPINAE
:/ ADD NAME=BVPINFP
:/ ADD NAME=BVPINFQ
:/ ADD NAME=BVPINGU
:/ ADD NAME=BVPINQJ
:/ ADD NAME=BVPINSL
:/ ADD NAME=BVPIPEP
:/ ADD NAME=BVPIPFQ
:/ ADD NAME=BVPIPIA
```

```
:/ ADD NAME=BVPISEP
:/ ADD NAME=BVPISOS
:/ ADD NAME=BVPJJND
:/ ADD NAME=BVPJRND
:/ ADD NAME=BVPLSND
:/ ADD NAME=BVPMB25
:/ ADD NAME=BVPMB30
:/ ADD NAME=BVPMEND
:/ ADD NAME=BVPMNDO
:/ ADD NAME=BVPMLIB
:/ ADD NAME=BVPPACG
:/ ADD NAME=BVPPACS
:/ ADD NAME=BVPPACX
:/ ADD NAME=BVPPAGX
:/ ADD NAME=BVPPC25
:/ ADD NAME=BVPPE25
:/ ADD NAME=BVPPG20
:/ ADD NAME=BVPPG25
:/ ADD NAME=BVPPJ25
:/ ADD NAME=BVPPPAF
:/ ADD NAME=BVPPP25
:/ ADD NAME=BVPPQCA
:/ ADD NAME=BVPPQCE
:/ ADD NAME=BVPPRGS
:/ ADD NAME=BVPREOR
:/ ADD NAME=BVPREST
:/ ADD NAME=BVPRESY
:/ ADD NAME=BVPRIND
:/ ADD NAME=BVPRMTD
:/ ADD NAME=BVPROAD
:/ ADD NAME=BVPRPND
:/ ADD NAME=BVPRPTY
:/ ADD NAME=BVPRP25
:/ ADD NAME=BVPRRND
:/ ADD NAME=BVPRSAD
:/ ADD NAME=BVPSADM
:/ ADD NAME=BVPSASY
:/ ADD NAME=BVPSAVE
:/ ADD NAME=BVPSIPM
:/ ADD NAME=BVPSMTD
:/ ADD NAME=BVPSTAT
:/ ADD NAME=BVPTRDU
:/ ADD NAME=BVPTRED
:/ ADD NAME=BVPTRJC
:/ ADD NAME=BVPTRPF
:/ ADD NAME=BVPTRRP
:/ ADD NAME=BVPTRUP
:/ ADD NAME=BVPTYND
:/ ADD NAME=BVPUBPM
:/ ADD NAME=BVPUPDP
:/ ADD NAME=BVPUPDT
:/ ADD NAME=BVPUPGP
:/ ADD NAME=BVPUPND
:/ ADD NAME=BVPUTMP
:/ ADD NAME=BVPUTM1
```

```
:/ ADD NAME=BVPUTM2
:/ ADD NAME=BVPUTU1
:/ ADD NAME=BVPUTU2
:/ ADD NAME=BVPUV25
:/ ADD NAME=BVPVING
:/ ADD NAME=BVPVINS
:/ ADD NAME=BVPXPAF
:/ ADD NAME=BVPXPDM
:/ ADD NAME=BVPYSMC
F+
//
```

Loading of Generation Skeleton Files

D06SKEL module: '$prfj.D6' job

Creation and loading of skeleton files.

Step	Program	Comments
STEP1	IDCAMS	DELETE/DEFINE SA SC SG SN SR SS SP
STEP2	IDCAMS	Loading (REPRO) of SA SC SG SN SR SS SP
STEP3	IDCAMS	DELETE of SF file
STEP4	IEBGENER	Loading of SF file

Execution JCL

```
//$PRFJ.D6  JOB ($CCPT),'PAC D06SKEL',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D06SKEL                              *
//*   LOADING OF PRODUCT SKELETON FILES FOR GENERATION                *
//*    .STEP1 : DELETE DEFINE SKELETON FILES SA SC SG SN SR SS SP     *
//*    .STEP2 : LOADING VSAM SKELETON FILES SA SC SG SN SR SS SP      *
//*    .STEP3 : DELETE "SF"                                           *
//*    .STEP4 : ALLOCATING AND LOADING "SF"                           *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DSN=$INDSN..BVPSY(DFBVPSA),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPSC),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPSG),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPSN),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPSR),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPSS),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPSP),DISP=SHR
//*
//STEP2  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
```

```
//SYSPRINT DD SYSOUT=$OUT
//SAO      DD DSN=$INDSV..BVPSA,DISP=SHR
//SCO      DD DSN=$INDSV..BVPSC,DISP=SHR
//SGO      DD DSN=$INDSV..BVPSG,DISP=SHR
//SNO      DD DSN=$INDSV..BVPSN,DISP=SHR
//SRO      DD DSN=$INDSV..BVPSR,DISP=SHR
//SSO      DD DSN=$INDSV..BVPSS,DISP=SHR
//SPO      DD DSN=$INDSV..BVPSP,DISP=SHR
//SAI      DD DSN=$HLQ..SBVPPF5(BVPSA$LSK),DISP=SHR
//SCI      DD DSN=$HLQ..SBVPPF6(BVPSC$LSK),DISP=SHR
//SGI      DD DSN=$HLQ..SBVPPF5(BVPSG$LSK),DISP=SHR
//SNI      DD DSN=$HLQ..SBVPPF5(BVPSN$LSK),DISP=SHR
//SRI      DD DSN=$HLQ..SBVPPF5(BVPSR$LSK),DISP=SHR
//SSI      DD DSN=$HLQ..SBVPPF5(BVPSS$LSK),DISP=SHR
//SPI      DD DSN=$HLQ..SBVPPF5(BVPSP),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (SAI)   OUTFILE (SAO)
  REPRO INFILE (SCI)   OUTFILE (SCO)
  REPRO INFILE (SGI)   OUTFILE (SGO)
  REPRO INFILE (SNI)   OUTFILE (SNO)
  REPRO INFILE (SRI)   OUTFILE (SRO)
  REPRO INFILE (SSI)   OUTFILE (SSO)
  REPRO INFILE (SPI)   OUTFILE (SPO)
//*
//STEP3  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..BVPSF)
//*
//STEP4  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=$HLQ..SBVPPF7(BVPSF),DISP=SHR
//SYSUT2   DD DSN=$INDSN..BVPSF,DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,(10,1)),
//         DCB=(RECFM=FB,LRECL=119,BLKSIZE=11900)
//*
//
```

Loading of Error messages and online help

D07AE0 module: '$prfj.D7' job

This JCL defines, and then loads the AE error messages file.

It must be executed every time a version is re-installed.

To take one or more languages into account, you must add as many parameter lines which contain the two-characters language code in position 3.

To take all the available languages into account, you must enter a parameter line which contains '**' in position 3.

The AE file must be closed in on-line mode.

Step	Program	Comments
INPUT	BVPTU001	Recognition of the language parameter
DELDEF	IDCAMS	DELETE DEFINE of AE
MAXKEY	IDCAMS	Maximum key of AE
PTUCAE	BVPTUCAE	Loading of AE

Execution JCL

```
//$PRFJ.D7  JOB ($CCPT),'PAC D07AE0',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D07AE                                *
//*********************************************************************
//INAE   EXEC BVPINAE
  **
/*
```

Installation of the Administration Database

Users File Initialization

D08INGU module: '$prfj.D8' job

This JCL should be submitted upon the first installation only.

It defines and then loads the GU file with the 'TEST' and 'ADMIN' user codes. These user codes will be used to activate the keys and to execute the test JCLs.

Note:

For a re-installation on a version higher than or equal to 3.0, if you want to restore the Administration Database of the previous version, you will first have to retrieve the GU file which corresponds to the PE backup.

Step	Program	Comments
STEP1	IDCAMS	DELETE DEFINE of GU
STEP2	PTUIGU	creation of data

Execution JCL

```
//$PRFJ.D8  JOB ($CCPT),'PAC D08INGU',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D08INGU                              *
//*         - INITIALIZATION OF USER FILE GU -                        *
//*                                                                   *
//*     EXECUTE THIS JOB ONLY FOR FIRST INSTALLATION                  *
//*                                                                   *
//*********************************************************************
//*
//INGU   EXEC BVPINGU
```

Initialization of Generation Data Groups

D08INIAD module: '$prfj.D8A' job

This JCL must be submitted upon the first installation only.

It creates GDG files and initializes the journal and Administration Database backups.

Caution

If the files are managed under SMS, delete the lines DD //GDGMOD from the JCL of the IDCAMS steps before submitting the job.

Step	Program	Comments
STEP1	IDCAMS	GDG of administration journal file
STEP2	IEBGENER	Initialization of this file (PK)
STEP3	IDCAMS	GDG of admin. backup file
STEP4	IEBGENER	Initialization of this file (PE)
STEP5	IDCAMS	GDG of backup file of SCM journal
STEP6	IEBGENER	Initialization of this file (JQ)

Execution JCL

```
//$PRFJ.D8A JOB ($CCPT),'PAC D08INIAD',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D08INIAD                             *
//* JOB TO RUN ONLY FOR THE 1ST INSTALL OF DATABASE ADMINISTRATOR     *
//*    . BUILDING OF INDEX DATA-GROUP FOR                             *
//*      "PE" SAVE FILE, "PK" ARCHIVAL FILE, "JQ" ARCHIVAL FILE       *
//*    . "PK" FILE INITIALIZATION                                     *
```

```
//*    . "JQ" FILE INITIALIZATION                                    *
//*    . LOADING OF TEST DATABASE ON "PE" FILE                       *
//*  ->NOTE                                                          *
//*    ----                                                          *
//*    IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS           *
//*****************************************************************
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDSN..BVPPK,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=FB,LRECL=170,BLKSIZE=6800)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDSN..BVPPK) LIMIT (3) SCR)
//*
//STEP2  EXEC PGM=IEBGENER
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=170,BLKSIZE=170)
//SYSUT2   DD DSN=$INDSN..BVPPK(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,1),
//         DCB=($DSCB,RECFM=FB,LRECL=170,BLKSIZE=6800)
//*
//STEP3  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDSN..BVPPE,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDSN..BVPPE) LIMIT (3) SCR)
//*
//STEP4  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=$HLQ..SBVPPF2(BVPPE),DISP=SHR
//SYSUT2   DD DSN=$INDSN..BVPPE(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,(220,10),RLSE),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//*
//STEP5  EXEC PGM=IDCAMS
```

```
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDSN..BVPJQ,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=FB,LRECL=1119,BLKSIZE=11190)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDSN..BVPJQ) LIMIT (3) SCR)
//*
//STEP6  EXEC PGM=IEBGENER
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=1119,BLKSIZE=1119)
//SYSUT2   DD DSN=$INDSN..BVPJQ(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITSN,
//         VOL=SER=$VOLSN,
//         SPACE=(TRK,1),
//         DCB=($DSCB,RECFM=FB,LRECL=1119,BLKSIZE=1119)
//
```

Loading of the Administration Database

D08RSAD module: '$prfj.D8B' job

This JCL must be submitted upon the first installation only.

It runs the RSAD procedure in order to restore the Administration Database with the backup file created in the previous '$prfj.D8A' job.

Note

If you have a Database of a previous version, consult the chapter dedicated to the retrieval of user parameters.

Caution

To use VisualAge Pacbase, you need an access key. At the end of this step, an evaluation key is installed, but this key only allows the execution of some procedures, in particular the installation procedures.

This access key must be updated via Administrator workbench, to allow an unrestricted access to VA Pacbase.

Execution JCL

```
//$PRFJ.D8B JOB ($CCPT),'PAC D08RSAD',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
```

```
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D08RSAD                              *
//* CREATION OF THE TEST DATABASE ADMINISTRATOR                       *
//* INDICATE THE LANGUAGE CODE OF THE SITE ON Y LINE COL 10-11        *
//*********************************************************************
//TESTRSAD  EXEC BVPRSAD
 *ADMIN   ADMIN
 Y     20EN
//
```

Initialization of the QJ archive file

D08TINQJ module: '$prfj.D8T' job

This JCL should be executed only upon the first installation.

It defines and then loads the QJ 'Archive journal of the SCM Tools Interface module'.

Step	Program	Comments
STEP1	IDCAMS	DELETE DEFINE QJ
PCMINI	PCMINI	Initialization of QJ

Execution JCL

```
//$PRFJ.D8T JOB ($CCPT),'PAC D08TINQJ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D08TINQJ                             *
//*         - INITIALIZATION OF SYSTEM FILE QJ -                      *
//*                                                                   *
//*     EXECUTE THIS JOB ONLY FOR FIRST INSTALLATION >= V300          *
//*                                                                   *
//*********************************************************************
//INQJ      EXEC BVPINQJ
```

Loading of the Administration Model

D08XMET module: '$prfj.8X' job

This JCL must be submitted upon the first installation and at each re-installation.

It is used to run the VING procedure and to install the administration model.

Execution JCL

```
//$PRFJ.8X  JOB ($CCPT),'PAC D08XMET',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*    - ADMINISTRATION DATABASE -                                    *
//*      MODEL INSTALLATION                                           *
//*      INDICATE THE FILE NECESSARY FOR ROAD PROCEDURE               *
//*********************************************************************
//VING   EXEC BVPVING,FDIC='$HLQ..SBVPPF8(BVPMETAD)',
//       USER=ADMIN
 *ADMIN   ADMIN   ***       VINSI
/*
//*VINS.PAC7MR  DD DSN=ROADFILE
```

Access Key Input

This step is performed with Administrator workbench and consists in executing the following operations:

- Access key entering,
- Targets definition,
- Key activation.

These operations are presented in the 'AD workbench User's Guide', 'Prior administration actions' chapter. They are detailed in the Online Help of Administrator workbench.

Backup of the Administration Database

Performing this step is recommended since it prevents you from having to enter the key again in case of a problem.

List of components with their date of creation

D99INSL module: '$prfj.D99' job

This job prints the following lists:

- A list of batch and on-line programs, with their generation characteristics,
- A list of permanent system files AE ,SA, SC, SG, SR, SS, SN, SP and SF with the date of their creation.

In case of system operation problem, these lists should be printed in order to communicate to VisualAge Pacbase Support all the installation references.

This job also executes the INSL procedure.

Meaning of return code:

Return code	label
6	Load-module not found in the library
A	Not standard

Execution JCL

```
//$PRFJ.D99  JOB ($CCPT),'PAC D99INSL',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - D99INSL                              *
//*    THE FOLLOWING JOB PROVIDES THE LIST OF PROGRAMS                *
//*    AND SYSTEM FILES INSTALLED ON THE SITE.                        *
//*********************************************************************
//INSL   EXEC BVPINSL
/*
```

Additional Installations

Utility to purge work files

The JCLPURGE JCL is provided in standard.

Its purpose is to purge the WS workstation work file (its size is 150 cylinders, to allow voluminous requests), the PA PAF work file, and the LB job execution report file.

It is recommended to execute this JCL daily or to insert it in the CICS startup to prevent a possible problem in online mode.

SQL sources for access to DB2 catalog

With the B.....GN... choice, the system generates the SQL and calls the SQL update module: BVPRS12.

The BVPRS12 SQL update load-module is automatically provided by SMP/E in the hlq.SBVPSRC PDS. It just allows the online use but does not perform updates.

To benefit from the update functions, you must download the BVPRS12 source from the VisualAge Pacbase Support: http://www.ibm.com/software/awdtools/vapacbase/support.htm. You must adapt it and install it as any standard DB2 program:

- Pre-processor preparation, resulting in a COBOL source and a DBRM (same name as the COBOL source).

- Compilation and link-edit of the COBOL source, resulting in an executable program (BVPRS12).

The plan is built with the DBRM, using the TSO DB2I application (this being the BIND operation).

The CICS RCT table is updated with the development Database transaction code and the code of the plan previously built.

Special authorizations

The TSO userid used for the BIND must allow for the consultation of the following catalog tables:

- SYSIBM.SYSDATABASE
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSTABLES
- SYSIBM.SYSCOLUMNS

Notes:

- Updates are performed via SQL dynamic statements.
- DB2 tables or views which are accessed or updated are not qualified, i.E. the user's CICS Userid will implicitly qualify the DB2 object's name.

CICS Complements for EXCI

This step is to be executed only on the sites which use SCM.

You will have to check that the EXCI interface for the communication with the CICS system is operational.

If this is not the case, you will have to make sure that the CICS is started with the ISC=YES option and you will have to define a "SESSION/CONNECTION" set by executing the EC00CSD JCL as follows:

Execution JCL

```
//$PRFJ.E0 JOB ($CCPT),'SYSTEM DFHCEXCI',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - CICSCSD                              *
//*           EXCI: CSD BATCH UPDATE SYSTEM                           *
//*********************************************************************
//DFHCSDUP EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=$CSDL,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DFHCSD   DD DSN=$DFHCSD,DISP=SHR
//SYSIN    DD *
```

```
************************************************************
DEFINE CONNECTION($EXCN) GROUP($GROUE)
       ACCESSMETHOD(IRC) PROTOCOL(EXCI) CONNTYPE(GENERIC)
       SINGLESESS(NO) DATASTREAM(USER) RECORDFORMAT(U)
       QUEUELIMIT(NO) MAXQTIME(NO) AUTOCONNECT(NO)
       INSERVICE(YES) ATTACHSEC(LOCAL)
       BINDSECURITY(NO) USEDFLTUSER(NO) XLNACTION(KEEP)
DEFINE SESSIONS($EXSS) GROUP($GROUE)
       CONNECTION($EXCN) PROTOCOL(EXCI) MAXIMUM(0,0)
       RECEIVEPFX($EXPF) RECEIVECOUNT(5) SENDSIZE(4096)
       RECEIVESIZE(4096) SESSPRIORITY(0) AUTOCONNECT(NO)
       BUILDCHAIN(YES) USERAREALEN(0) IOAREALEN(4096,4096)
       RELREQ(NO) DISCREQ(NO) NEPCLASS(0)
       RECOVOPTION(SYSDEFAULT)
************************************************************
ADD GROUP($GROUE) LIST($LIST)
```

Security Systems Interface

Introduction

Security systems provide a mechanism for data access control. They perform user identification and verification, and they control resource access authorizations.

The Security System Interface is designed to ensure the communication of these controls between the security system installed on site and VA Pac.

In order to ensure a standardized control for all security systems, VA Pac is not directly interfaced with the security system, but with SAF (System Authorization Facility), via the RACROUTE macro-instructions for RACF or the TSS macro-instructions for TOPSECRET.

To enable the communication between the security system and VA Pac, two types of operations are required: the first ones must be performed in the Administrator workbench module, and the others in the security system.

- From the Administrator workbench module

  You must change the access key, define the type of security system used (RACF or TOPSECRET) and the class in which VA Pac logical resources are defined.

  For more information, consult the Administrator & Developer workbench User's Guide , Chapter "Prior Administration Actions ", and the online help of Administrator workbench, on the "Access Key" and "Security" browsers.

- From the security system

  You must perform the following operations:

  – create a RACF or TOPSECRET resource class in accordance with the security system chosen.

To allow the security system to work properly, all the VA Pac logical resources are associated with a RACF or TOPSECRET class.

The VA Pac logical resources correspond to all the possible access authorizations for each library.

– create resources in RACF or TOPSECRET.

  This operation consists in defining, in the RACF or TOPSECRET tables, the access authorizations to the VA Pac libraries if you want a complete control of resources from RACF or TOPSECRET.

– specify the user access authorizations to the resources either by using commands, or by installing sub-programs.

Finally, to use sockets (specific to CICS), you must install a CICS listener which does not standardly include an authentication system. The security system performs the control via the BVPCICSE security exit-user provided by SMP/E in the hlq.SBVPMTR8 PDS.

The BVPCICSE exit-user must be declared in the CSD with the 'EXECKEY(CICS)' clause (see the D00CSD installation JCL) in the following way:

In CICS, enter the 'EZAC AL' command:

```
EZAC,ALter
ENTER ONE OF THE FOLLOWING
CICS         ===>                      Enter Yes!No
LISTENER     ===>  Y                   Enter Yes!No
```

Select the listener and enter:

```
EZAC,ALter,LISTENER
ENTER ALL FIELDS
APPLID       ===> A6ECCSXP     APPLID of CICS System
NAME         ===> CSKL         TRANSACTION NAME OF LISTENER
```

The following screen is then displayed. You must indicate code of the exit-user:

```
EZAC,ALter,LISTENER
OVERTYPE TO ENTER
APPLID       ===> A6ECCSXP     APPLID of CICS System
TRANID       ===> CSKL         Transaction Name of Listener
PORT         ===> 09957        Port Number of Listener
IMMEDIATE    ===> YES          Immediate Startup Yes!No
BACKLOG      ===> 040          Backlog Value for Listener
NUMSOCK      ===> 100          Number of Sockets in Listener
```

```
MINMSGL      ===> 004          Minimum Message Length
ACCTIME      ===> 030          Timeout Value for ACCEPT
GIVTIME      ===> 010          Timeout Value for GIVESOCKE
REATIME      ===> 300          Timeout Value for READ
FASTRD       ===> YES          Read Immediately    Yes!No
TRANTRN      ===> YES          Translate TRNID     Yes!No
TRANUSR      ===> YES          Translate User Data Yes!No
SECEXIT      ===> BVPCICSE     Name of Security Exit
WLM groups   ===> CICSSEXP     ===>              ===>
```

Implementation for RACF

If RACF is used, you cannot work in VisualAge Pacbase with a user code different from the one used to connect to CICS or TSO.

There are always two control levels:

- the VA Pac user is the user of the system (RACF)
- the resources (Library-Authorization level) are managed in RACF.

CREATION OF RESOURCES

Resources are created with the 'RDEFINE' procedure.

DEFINITION OF RESOURCES

- Definition of the resource class

  The resource class is defined via the ICHERCDE and ICHRFRTB macros:

  cccc ICHERCDE CLASS=cccc, ID=128, MAXLNTH=4, FIRST=ALPHANUM, OTHER=ANY, POSIT=25, OPER=NO

  cccc ICHRFRTB CLASS=cccc,ACTION=RACF

  with cccc as the code of the RACF resource class dedicated to VA Pac.
- Definition of the access authorizations to the resources

  These authorizations must be assigned using PERMIT as follows:

  PERMIT 4$P CLASS(cccc) ID(user) ACCESS(ALTER) for VA Pac Administrators

  PERMIT nLLL CLASS(cccc) ID(user) ACCESS(ALTER), n varies from 0 to 3 and corresponds to the VA Pac authorization level on the LLL library.

ONLINE MODE REQUESTS

In online mode, a command allows to execute a RACF request to determine the user access authorizations to the resources ('EXEC QUERY SECURITY' in the BVPRACF control sub-program).

- Connection in TUI mode:

  The 'user' and 'password' fields are locked.

  They are retrieved by an 'EXEC CICS ASSIGN' upon the connection to CICS (CESN transaction or equivalent).

  All authorization controls are made for this user via the RACF request.
- Connection via the workbench:

  Only the middleware user and password are to be entered.

  As for the connection in TUI mode, the user code is retrieved by an 'EXEC CICS ASSIGN'.

BATCH REQUESTS

Contrary to the on-line mode, there is no command you could use to execute a RACF request: therefore, you must execute the RACF request via an assembler sub-program.

This BVPSECUR sub-program is provided by SMP/E in the hlq.SBVPMBR8 PDS.

Moreover, you must install a SVC in the LPA library.

To do so, you must:

1. Declare the SVC with the appropriate routine number.

Example for number 232: in SYS1.PARMLIB(IEASVC00), add the line SVCPARM 232,REPLACE,TYPE(3),EPNAME(IGC0023B) where IGC0023B is the name of the SVC load module.

2. Link the BVPSECUR object module to create the name of the load module.

3. Add this load module name in the SVC load system library.

4. ZAP the BVPSECB VA Pac module with the chosen SVC routine number.

5. After the LPA library update, you must execute an IPL CLPA to take the modifications into account.

The E10RACF sample JCL, that is supplied with the installation JCLs, performs steps 2 and 3, and the E11RACF JCL executes step 4.

Then, you must check whether the user is authorized to run the procedure. In case of a multi-user procedure (UPDT for instance), each identification line triggers the control of the user's authorization on the library.

Implementation for TOPSECRET

The sources of the sub-programs (assembler) which allow the access to the TOPSECRET tables are supplied by SMP/E in the hlq.SBVPSRC PDS and must be compiled.

COMPILATION OF THE ACCESS SUB-PROGRAMS

For the compilation of BVPTSS and BVPTSSC, the ″OPMAT″ TSS library must be specified in the SYSLIB line of the Assembler compilation program.

BVPTSSC is a CICS program. It must be translated before being compiled and link-edited.

BVPTSSC and the TSSCAI program (Computer Associates) must be declared in the CICS CDS. They must be loaded in one of the DFHRPL's load-module library .

CREATION OF RESOURCES

TSS ADD(dept-name) cccc(nlib) cccc(nlib) ...

dept-name = department name

n = authorization level

lib = Library code

cccc = Resource class

EXAMPLE

Let a Database, with two Libraries BI1 and BI2. The resource class will be, for access authorizations to Libraries:

```
4BI1 3BI1 2BI1 1BI1 0BI1 4BI2 3BI2 2BI2 1BI2 0BI2
```

Note: there is no difference between the global authorization and the database authorization, for this latter does not exist in the Security System Interface.

The resources corresponding to the general access authorization must be defined with a special Library code $$$:

4$$$ 3$$$ 2$$$ 1$$$ 0

Since '*' is the generic character for the security system, the Inter-Library '***' mode must be coded as '£££' (or '###' depending on the keyboard used).

There is another special library code $P, which is used to declare an administrator.

DEFINITION OF RESOURCES

- Definition of the resource class

  SS ADD (RDT) RESCLASS(cccc) RESCODE(xx)

  cccc = code of the resource class which corresponds to VA Pac

  xx = hexadecimal code indicating the type of resource

- Definition of access authorizations

  TSS PERMIT(user-code) cccc(nlib)

  TSS PERMIT(user-code) cccc(nlib)

VA Pacbase/Endevor TSO interface

Warning

The system preparation, installation and operation use some standard functions of the operating system and of VSAM. The user is entirely responsible for any adaptation which modifies the parameters entered upon installation (described in chapter 3).

In particular, all modifications on JCLs and executable modules' names etc., must be performed with caution. This kind of modifications often causes abnormal conditions which are sometimes difficult to detect.

In case of an incident during the execution of an installation phase, the phase must be started again from the beginning without any JCL modification of the specific JOBs.

Allocation of processors to the PROCESSOR GROUP

Reminder: The VA Pacbase Endevor system is constituted of two types of elements:

- 'VA Pac' type: All VA Pac entities (compilable or not) are stored in Endevor Types called 'VA Pac' types. Several 'VA Pac' types can be created.
- 'INFOPAC' type: Each VA Pac type element is associated with an 'INFOPAC' type, which is in its turn associated with this 'VA Pac' type. The 'INFOPAC' type is transparent to the user. It cannot be modified, except on some specified processors.

Three PROCESSOR GROUPs must be defined for the VA Pacbase Endevor system.

Two of these PROCESSOR GROUPs are allocated to each 'VA Pac' type and one is allocated to each 'INFOPAC' type.

These PROCESSOR GROUPs must be defined in the following way:

- Non-compilable VA Pac entities ('VA Pac' type)
  - PROCESSOR GROUP: (user-defined name)
  - GENERATE PROCESSOR: (5)
  - DELETE PROCESSOR: (2)
  - MOVE PROCESSOR: (3)
- Compilable VA Pac entities ('VA Pac' type)
  - PROCESSOR GROUP: (user-defined name)
  - GENERATE PROCESSOR: (1)
  - DELETE PROCESSOR: (2)
  - MOVE PROCESSOR: (3)
- Compilable or non-compilable VA Pac entities ('INFOPAC' type)
  - PROCESSOR GROUP: (user-defined name)
  - GENERATE PROCESSOR: (4)
  - DELETE PROCESSOR: *NOPROC*
  - MOVE PROCESSOR: *NOPROC*

Where:

(1): 'PRCSGENP' member of the 'BVPSY' parameters PDS.

This processor compiles and link-edits the generated VA Pac entity, and creates its FOOTPRINT.

Note: This JCL must be adapted to the site standards before it is installed in the Endevor system; in particular the LOADTEST parameter must correspond to the user load-modules library.

(2): 'PRCSDELP' member of the 'BVPSY' parameters PDS.

This processor executes the DELETE action on the INFOPAC type associated with the VA Pac type (See the 'TYND' batch procedure). It also communicates to EXIT3 the VA Pac data of the element to be deleted. This data is contained in its homonym, but whose type is INFOPAC. EXIT3 then relates this data to the data of the Endevor context and prepares the VA Pac Database update transactions.

(3): 'PRCSMOVP' member of the 'BVPSY' parameters PDS.

This processor executes the MOVE action on the INFOPAC type associated with the VA Pac type: it transfers it from STAGE 1 to STAGE 2 (See 'TYND' batch procedure). It also communicates to EXIT3 the VA Pac data of the

element to be transferred. This data is contained in its homonym, but whose type is INFOPAC. EXIT3 then relates this data to the data of the Endevor context and prepares VA Pac Database update transactions.

(4): 'PRCSGENI' member of the 'BVPSY' parameters PDS.

This processor executes the ADD action on the INFOPAC type associated with the VA Pac type. IT also modifies the compiled load module FOOTPRINT (when the entity is compilable) in order to allocate the FOOTPRINT of the current INFOPAC to it. It finally communicates to EXIT3 the VA Pac data of the element to be transferred. This data is contained in its homonym, but whose type is INFOPAC. EXIT3 then relates this data to the data of the Endevor context and prepares VA Pac Database update transactions.

Note: The LOADTEST parameter must correspond to the user load-modules library.

(5): 'PRCSGEPP' member of the 'BVPSY' parameters PDS.

This 'GENERATE' type processor is only used for the 'TRANSFER' action of a non-compilable VA Pac element. It generates and executes the 'TRANSFER' action of the 'INFOPAC' element.

Important: All these processors can be used without modification. They can also be customized to meet site's requirements. Proceed with caution and carefully follow the recommendations given in this manual (see Chapter 'Description of the Interface Elements', Subchapter 'Interface Processors' of the 'Endevor Reference Manual').

Installation of the system

Execute the six following JCLs:

- Zap on EXIT2 'C1UEXT02'.
- Loading of the system parameters' PDS.
- Initial preparation of files.
- Loading of EXITs in an authorized library
- Loading of TSO / VA Pacbase Endevor messages into the site ENDEVOR library.
- Retrieval from the 2.5 version (before executing this JOB, read carefully the "Retrieval" chapter in the "VA Pacbase Endevor TSO Reference Manual").

EXIT2 ZAP 'C1UEXT02'

D11ZXIT module: '$prfj.ZXIT' job

This job contains the following steps:

Step	Program	Comments
STEP01	IMASPZAP	EXIT2 ZAP. It concerns the member name of the VA Pacbase Endevor messages 'ISPMLIB' library. It must be executed only if the value chosen for the $MSGSXH parameter is other than 50.

Execution JCL

```
//$PRFJ.XIT JOB ($CCPT),'ZAP EXIT USER',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*************************************************
//*         --- ZAP 'C1UEXT02' EXIT USER ---      *
//*************************************************
//*
//*-------------------------------------------------*
//*               I M P O R T A N T                 *
//*                  -----------                    *
//*  IT IS NOT NECESSARY TO EXECUTE THIS JOB        *
//*     IF THE SUFFIX OF THE PACBASE USER MESSAGES  *
//*     MEMBER OF ENDEVOR 'ISPMLIB' LIBRARY CONTAINS *
//*     THE VALUE '50' (CHECK 'MSGSX' PARAMETER).   *
//*  IF THE SUFFIX IS MODIFIED, THIS STEP MUST BE   *
//*     EXECUTED. IN ORDER TO DO SO, THE 'MSGSHX'   *
//*     MUST BE CODED WITH THE CORRESPONDING HEXA-  *
//*     DECIMAL VALUE OF THE NEW SUFFIX.            *
//*     FOR INST: IF THE NEW SUFFIX IS: MSGSX='89', *
//*               THE 'MSGSXH' PARAMETER MUST BE    *
//*               CODED: MSGSXH='F8F9'.             *
//*-------------------------------------------------*
//*
//STEP01  EXEC PGM=IMASPZAP
//SYSLIB   DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
 NAME C1UEXT02 C1UEXT02
 VER 0303 F5F0
 REP 0303 $MSGSXH
//*
//
```

Loading of the system parameters' PDS

D12SY module: '$prfj.SY' job

This job contains an IEBUPDTE of the following members:

- Input dedicated to SORT utilities
- Definitions (DELETE/DEFINE), verifications (VERIFY) and REPRO of VSAM files

- BLBVPQU member which contains the building command of the 'QU' backup index of the 'UQ' VA Pac update file.
- 'ISPLOGON' member which contains the 3 VSAM files used by the VA Pacbase Endevor system. They must be defined in the TSO procedure of the user LOGON.
- 'ISPF' member which contains the allocation of the Exits trace file. Since its contents are specific to each user, it must be copied in the ISPF connection procedure.
- 'CIUU$msgsx' member (see the meaning of '$MSGSX' parameter in the table of parameters). It contains the TSO messages of the VA Pacbase Endevor system and is copied in the Endevor 'ISPMLIB' library.
- 'PRCSGENI' member: JCL of the 'GENERATE' type processor for 'INFOPAC'-type ELEMENTS (1)
- 'PRCSGENP' member: JCL of the 'GENERATE' type processor for 'VA Pac'-type ELEMENTS with compilation (1)
- 'PRCSGEPP' member: JCL of the 'GENERATE' type processor for 'VA Pac'-type ELEMENTS without compilation, related to the 'TRANSFER' action (1).
- 'PRCSDELP' member: JCL of the 'DELETE' type processor for 'VA Pac'-type ELEMENTS (see PHASE 1)
- 'PRCSMOVP' member: JCL of the 'MOVE' type processor for 'VA Pac'-type ELEMENTS (1)

(1) 'see Allocation of processors to the PROCESSOR GROUP'

Execution JCL

```
//$PRFJ.SY JOB ($CCPT),'LOAD PARAM.',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*
//*************************************************************
//*       LOADING PACBASE-ENDEVOR PARAMETERS IN 'SY' PDS       *
//*************************************************************
//*
//*    *-------------------------------------------------*
//*    *               I M P O R T A N T                 *
//*    *                  -----------                    *
//*    *      BEFORE EXECUTING THIS JOB, REPLACE ALL     *
//*    *          ':/' BY './' VIA THE EDITOR.           *
//*    *-------------------------------------------------*
//*
//*
//ET010   EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT1   DD DSN=$INDSN..BVPSY,DISP=SHR
//SYSUT2   DD DSN=$INDSN..BVPSY,DISP=SHR
//SYSIN    DD DATA,DLM='F+'
:/ ADD NAME=DF$BASE.QJ
```

```
 DELETE ($INDSVE..$BASE.QJ) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSVE..$BASE.QJ)          -
                  SHR (2)         NUMBERED          -
                  VOL ($VOLSV)    REC (310)         -
                  RECSZ (310,310) RUS )             -
        DATA    ( NAME ($INDSVE..$BASE.QJ.D)        -
                  CISZ (4096) )  /*: CATALOG ($CATU) */
:/ ADD NAME=DF$BASE.TS
 DELETE ($INDSVE..$BASE.TS) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSVE..$BASE.TS)          -
                  SHR (2 3)       KEYS (14 1)       -
                  VOL ($VOLSV)    REC (100)         -
                  RECSZ (080,080) RUS )             -
        INDEX   ( NAME ($INDSVE..$BASE.TS.I)        -
                 CISZ (4096) )                      -
        DATA    ( NAME ($INDSVE..$BASE.TS.D)        -
                  FSPC (10,5)                       -
                  CISZ (4096) )  /*: CATALOG ($CATU) */
:/ ADD NAME=DFBVPTY
 DELETE ($INDSVE..BVPTY) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSVE..BVPTY)             -
                  SHR (2,3)       KEYS (8,1)        -
                  REC  (150)                        -
                  VOL ($VOLSV)                      -
                  RECSZ (080,080) RUS )             -
        INDEX   ( NAME ($INDSVE..BVPTY.I)           -
                  CISZ (4096) )                     -
        DATA    ( NAME ($INDSVE..BVPTY.D)           -
                  FSPC (10,5)                       -
                  CISZ (4096) )  /*: CATALOG ($CATU) */
:/ ADD NAME=DFBVPUP
 DELETE ($INDSVE..BVPUP) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSVE..BVPUP)             -
                  SHR (2,3)       KEYS (43,0)       -
                  REC  (10000)                      -
                  VOL ($VOLSV)                      -
                  RECSZ (112,112) RUS )             -
        INDEX   ( NAME ($INDSVE..BVPUP.I)           -
                  CISZ (4096) )                     -
        DATA    ( NAME ($INDSVE..BVPUP.D)           -
                  FSPC (10,5)                       -
                  CISZ (4096) )  /*: CATALOG ($CATU) */
:/ ADD NAME=DFBVPUQ
 DELETE ($INDSVE..BVPUQ) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSVE..BVPUQ)             -
                  SHR (2,3)       NUMBERED          -
                  REC  (10000)                      -
                  VOL ($VOLSV)                      -
                  RECSZ (170,170) RUS )             -
        DATA    ( NAME ($INDSVE..BVPUQ.D)           -
                  CISZ (4096) )  /*: CATALOG ($CATU) */
:/ ADD NAME=DFBVPUU
 DELETE ($INDSVE..BVPUU) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSVE..BVPUU)             -
                  SHR (2,3)       NUMBERED          -
```

```
                  REC  (10000)                      -
                  VOL ($VOLSV)                      -
                  RECSZ (170,170) RUS )             -
        DATA    ( NAME ($INDSVE..BVPUU.D)           -
                  CISZ (4096) )  /*: CATALOG ($CATU) */
:/ ADD NAME=DLBVPUU
 DELETE ($INDSVE..BVPUU) CLUSTER
:/ ADD NAME=DL$BASE.QJ
 DELETE ($INDSVE..$BASE.QJ) CLUSTER
:/ ADD NAME=VERIFTY
 VERIFY FILE (PAC7TY)
:/ ADD NAME=BLBVPQU
   DEFINE GENERATIONDATAGROUP -
          (NAME ($INDUNE..BVPQU) LIMIT (3) SCR)
:/ ADD NAME=ISPLOGON
//*
//* THE FOLLOWING THREE FILES MUST BE DECLARED
//* INTO THE USER TSO LOGON PROCEDURE.
//*
//IKJ     EXEC PGM=IKJEFT01, ...
  .
  .
//PAC7TY   DD DSN=$INDSVE..BVPTY,DISP=SHR
//PAC7UP   DD DSN=$INDSVE..BVPUP,DISP=SHR
//PAC7UQ   DD DSN=$INDSVE..BVPUQ,DISP=SHR
  .
  .
//
:/ ADD NAME=ISPF
//*
//* EXITS TRACE FILE ALLOCATION MUST BE DECLARED
//* INTO THE USER ISPF CONNECTION PROCEDURE.
//*
CONTROL MAIN NOMSG
SET &PTRACE = &STR('$INDUNE..&SYSUID..PTRACE')
FREE  FI(PTRACE)
ALLOC FI(PTRACE) SHR REU DA(&PTRACE)
IF &LASTCC > 0 THEN DO
   FREE ATTRLIST(DCBTRACE)
   ATTR DCBTRACE  RECFM(F B) LRECL(120) BLKSIZE(12000) DSORG(PS)
   ALLOC FI(PTRACE) DA(&PTRACE) SPACE(20,10) TRACKS USING(DCBTRACE)
   FREE  FI(PTRACE)
   ALLOC FI(PTRACE) SHR REU DA(&PTRACE)
END
:/ ADD NAME=CIUU$MSGSX
CIUU$MSGSX.1E     'INVALID TYPE (INFOPAC)  ' .ALARM = YES .HELP = *
'''INFOPAC'' TYPE ELEMENTS ARE NOT ALLOWED FOR UPDATING
CIUU$MSGSX.2E     'INVALID ACTION (PACBASE)' .ALARM = YES .HELP = *
'ACTIONS ''ADD & UPDATE'' ARE NOT ALLOWED FOR PACBASE OUT OF PROCESSOR
CIUU$MSGSX.3E     'INVALID ACTION (PACBASE)' .ALARM = YES .HELP = *
'ONLY ACTIONS ''ADD & UPDATE'' ARE ALLOWED FOR PACBASE THROUGH PROCESSO
:/ ADD NAME=PRCSGENI
//*
//*--------------------------------------------------------------------*
//*  G E N E R A T E     P R O C E S S O R           (INFOPAC TYPE)  *
```

```
//*-------------------------------------------------------------------*
//*
//GENI    PROC ROOTPGM='BV',
//             STEPLIB='$HLQ..SBVPMBR8',
//             OUT='$OUT',
//             TY='$INDSVE..BVPTY',
//             UP='$INDSVE..BVPUP',
//             LOADTEST='???'
//*
//GEN00   EXEC PGM=BC1PDSIN,MAXRC=0,
//        EXECIF=(&C1COMMENT(36,5),NE,'*RND*')
//GEN02A   DD DSN=&&SYSOUT1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//GEN02B   DD DSN=&&SYSDMP1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//GEN03A   DD DSN=&&SYSOUT2,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//GEN04A   DD DSN=&&SYSOUT3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//GEN04B   DD DSN=&&SYSDMP3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//*
//GEN0A   EXEC PGM=IEBCOPY,MAXRC=4,
//        EXECIF=(&C1COMMENT(36,5),NE,'*RND*')
//SYSPRINT DD SYSOUT=&OUT
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,30)
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,30)
//IN       DD DSN=&LOADTEST,DISP=SHR
//OUT      DD DSN=&&TEMP,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=U,BLKSIZE=6144),
//            SPACE=(TRK,(10,10,10),RLSE)
//SYSIN    DD *
 C O=OUT,I=IN
 S M=((&C1ELEMENT,,R))
//*
//GEN0B   EXEC PGM=BSTCOPY,MAXRC=0,COND=(00,NE,GEN0A),
//        EXECIF=(&C1COMMENT(36,5),NE,'*RND*')
//SYSPRINT DD SYSOUT=&OUT
//IN       DD DSN=&&TEMP,DISP=(OLD,DELETE)
//OUT      DD DSN=&LOADTEST,DISP=SHR,FOOTPRNT=CREATE
//SYSIN    DD *
 C O=OUT,I=IN
 S M=((&C1ELEMENT,,R))
//*
//GEN01   EXEC PGM=&ROOTPGM.PNTRAN,
//        EXECIF=(&C1COMMENT(36,5),NE,'*RND*'),
//             PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_)I)
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
```

```
//PAC7PU   DD DSN=&UP,DISP=SHR
//PAC7TR   DD DSN=&&PAC7TR,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=58,BLKSIZE=58),
//            SPACE=(TRK,(01,01),RLSE)
//*
//GEN02   EXEC PGM=&ROOTPGM.PNPR10,MAXRC=0,
//        EXECIF=(&C1COMMENT(36,5),NE,'*RND*'),
//             PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_))
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSIN    DD DSN=&&PAC7TR,DISP=(OLD,DELETE)
//PAC7TZ   DD DSN=&TY,DISP=SHR
//PAC7BS   DD DUMMY,DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080)
//PAC7CW   DD DSN=&&CONWIN,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7EV   DD DSN=&&PAC7EV,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=126,BLKSIZE=12600),
//            SPACE=(TRK,(01,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT1,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP1,DISP=(MOD,PASS)
//*
//*-------------------------------------------------*
//*  PRINT 'INFOPAC' MEMBER OF THE ORIGINAL STAGE   *
//*  INTO SEQUENTIAL FILE 'PAC7IP'                  *
//*-------------------------------------------------*
//*
//GEN03   EXEC PGM=CONWRITE,MAXRC=0,
//        EXECIF=(&C1COMMENT(36,5),NE,'*RND*')
//CONWIN   DD DSN=&&CONWIN,DISP=(OLD,DELETE)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(02,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT2,DISP=(MOD,PASS)
//*
//*-------------------------------------------------*
//*  COPY SEQUENTIAL 'PAC7IP' IN THE VSAM 'PAC7UP'  *
//*  DEFINED IN TSO WITH A NEW DDNAME FOR EXIT3     *
//*-------------------------------------------------*
//*
//GEN04   EXEC PGM=&ROOTPGM.PNPR11,COND=(00,NE,GEN02),
//        EXECIF=(&C1COMMENT(36,5),NE,'*RND*')
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7EV   DD DSN=&&PAC7EV,DISP=(OLD,PASS)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(OLD,PASS)
//PAC7PU   DD DSN=&UP,DISP=SHR
//SYSOUT   DD DSN=&&SYSOUT3,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP3,DISP=(MOD,PASS)
//*
//*-------------------------------------------------*
//*           PRINT SYSOUTS, SYSUDUMP, ...          *
//*-------------------------------------------------*
//*
//GEN05   EXEC PGM=CONLIST,PARM=PRINT,COND=EVEN,MAXRC=0,
```

```
//        EXECIF=((&C1COMMENT(36,5),NE,'*IBM*'),
//               (&C1COMMENT(36,5),NE,'*RND*'))
//C1BANNER DD DSN=&&BANNER,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//C1PRINT  DD SYSOUT=&OUT,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171,DSORG=PS)
//LIST01   DD DSN=&&SYSOUT1,DISP=(OLD,DELETE)
//LIST02   DD DSN=&&SYSDMP1,DISP=(OLD,DELETE)
//LIST03   DD DSN=&&SYSOUT2,DISP=(OLD,DELETE)
//LIST04   DD DSN=&&SYSOUT3,DISP=(OLD,DELETE)
//LIST05   DD DSN=&&SYSDMP3,DISP=(OLD,DELETE)
//*
:/ ADD NAME=PRCSGEPP
//*
//*--------------------------------------------------------------------*
//*  G E N E R A T E    P R O C E S S O R            (PACBASE TYPE)  *
//*--------------------------------------------------------------------*
//*
//GEPP    PROC ROOTPGM='BV',
//             STEPLIB='$HLQ..SBVPMBR8',
//             OUT='$OUT',
//             TY='$INDSVE..BVPTY',
//             UP='$INDSVE..BVPUP'
//*
//TRS01   EXEC PGM=BC1PDSIN,MAXRC=0,
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE'))
//TRS03A   DD DSN=&&SYSOUT1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS03B   DD DSN=&&SYSDMP1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS04A   DD DSN=&&SYSOUT2,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS05A   DD DSN=&&SYSOUT3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS05B   DD DSN=&&SYSDMP3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//*
//TRS02   EXEC PGM=&ROOTPGM.PNTRAN,
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE')),
//             PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_)G)
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7PU   DD DSN=&UP,DISP=SHR
//PAC7TR   DD DSN=&&PAC7TR,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=58,BLKSIZE=58),
```

```
//            SPACE=(TRK,(01,01),RLSE)
//*
//TRS03   EXEC PGM=&ROOTPGM.PNPR10,
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE')),
//             PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_))
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSIN    DD DSN=&&PAC7TR,DISP=(OLD,DELETE)
//PAC7TZ   DD DSN=&TY,DISP=SHR
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7CW   DD DSN=&&CONWIN,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7EV   DD DSN=&&PAC7EV,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=126,BLKSIZE=12600),
//            SPACE=(TRK,(01,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT1,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP1,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*  PRINT 'INFOPAC' MEMBER OF THE ORIGINAL STAGE     *
//*  INTO SEQUENTIAL FILE 'PAC7IP'                    *
//*-----------------------------------------------------*
//*
//TRS04   EXEC PGM=CONWRITE,MAXRC=0,
//        EXECIF=(&C1ACTION(1,8),EQ,'GENERATE')
//CONWIN   DD DSN=&&CONWIN,DISP=(OLD,DELETE)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(02,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT2,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*  COPY SEQUENTIAL 'PAC7IP' IN THE VSAM 'PAC7UP'    *
//*  DEFINED IN TSO WITH A NEW DDNAME FOR  EXIT3      *
//*-----------------------------------------------------*
//*
//TRS05   EXEC PGM=&ROOTPGM.PNPR11,COND=(00,NE,TRS03),
//        EXECIF=(&C1ACTION(1,8),EQ,'GENERATE')
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7EV   DD DSN=&&PAC7EV,DISP=(OLD,PASS)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(OLD,PASS)
//PAC7PU   DD DSN=&UP,DISP=SHR
//SYSOUT   DD DSN=&&SYSOUT3,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP3,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*         TRANSFER 'INFOPAC' ELEMENT                  *
//*-----------------------------------------------------*
//*
```

```
//TRS06   EXEC PGM=C1BM3000,PARM=(PAC7BS,CXMSGSX),
//        EXECIF=(&C1ACTION(1,8),EQ,'TRANSFER')
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(OLD,PASS)
//CXMSGSX  DD SYSOUT=&OUT
//*
//*-------------------------------------------------*
//*           PRINT SYSOUTS, SYSUDUMP, ...          *
//*-------------------------------------------------*
//*
//TRS07   EXEC PGM=CONLIST,PARM=PRINT,COND=EVEN,MAXRC=0,
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE'))
//C1BANNER DD DSN=&&BANNER,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//C1PRINT  DD SYSOUT=&OUT,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171,DSORG=PS)
//LIST01   DD DSN=&&SYSOUT1,DISP=(OLD,DELETE)
//LIST02   DD DSN=&&SYSDMP1,DISP=(OLD,DELETE)
//LIST03   DD DSN=&&SYSOUT2,DISP=(OLD,DELETE)
//LIST04   DD DSN=&&SYSOUT3,DISP=(OLD,DELETE)
//LIST05   DD DSN=&&SYSDMP3,DISP=(OLD,DELETE)
//*
:/ ADD NAME=PRCSGENP
//*
//*-------------------------------------------------------------------*
//*  G E N E R A T E  (LOAD)  P R O C E S S O R      (PACBASE TYPE)  *
//*-------------------------------------------------------------------*
//*
//GENP    PROC ROOTPGM='BV',
//             STEPLIB='$HLQ..SBVPMBR8',
//             OUT='$OUT',
//             TY='$INDSVE..BVPTY',
//             UP='$INDSVE..BVPUP',
//             LOADTEST='???'
//*
//CNWRITE EXEC PGM=CONWRITE
//ELMSRC   DD DSN=&&ELMSRC,UNIT=SYSDA,DISP=(,PASS),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),
//            SPACE=(TRK,(15,15),RLSE)
//CONWIN   DD *
  WRITE ELEMENT          &C1ELEMENT
        FROM ENVIRONMENT &C1ENVMNT
             SYSTEM      &C1SYSTEM
             SUBSYSTEM   &C1SUBSYS
             TYPE        &C1ELTYPE
             STAGE       &C1STGID
        TO   DDN         ELMSRC.
//*
//COB    EXEC PGM=IGYCRCTL,MAXRC=04,
//            PARM='NOTERM,NOLIST,NOVBREF,NOOPT,NOTEST,NOXREF,APOST'
//*TEPLIB  DD DSNAME=CEE.SCEERUN,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
```

```
//SYSUT4   DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSUT5   DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSUT6   DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSUT7   DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSIN    DD DSN=&&ELMSRC,DISP=(OLD,DELETE)
//SYSLIN   DD DSN=&&LOADSET,UNIT=SYSDA,DISP=(MOD,PASS),
//            SPACE=(400,(500,200)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//SYSPRINT DD SYSOUT=&OUT
//*
//LKED    EXEC PGM=HEWL,PARM=(LIST,LET,XREF),MAXRC=04
//**      COND=(4,LT,COB)
//SYSLIB   DD DSNAME=CEE.SCEELKED,DISP=SHR
//         DD DSN=&LOADTEST,DISP=SHR
//SYSLIN   DD DSNAME=&&LOADSET,DISP=(OLD,DELETE)
//         DD *,DCB=BLKSIZE=80
 NAME &C1ELEMENT(R)
//SYSLMOD  DD DSN=&LOADTEST,DISP=SHR,FOOTPRNT=CREATE
//SYSUT1   DD UNIT=SYSDA,SPACE=(1024,(50,20))
//SYSPRINT DD SYSOUT=&OUT
//*
//TRS00   EXEC PGM=BC1PDSIN,MAXRC=0,
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE'))
//TRS02A   DD DSN=&&SYSOUT1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS02B   DD DSN=&&SYSDMP1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS03A   DD DSN=&&SYSOUT2,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS04A   DD DSN=&&SYSOUT3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//TRS04B   DD DSN=&&SYSDMP3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//*
//TRS01   EXEC PGM=&ROOTPGM.PNTRAN,
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE')),
//        PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_)G)
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7PU   DD DSN=&UP,DISP=SHR
//PAC7TR   DD DSN=&&PAC7TR,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=58,BLKSIZE=58),
//            SPACE=(TRK,(01,01),RLSE)
//*
//TRS02   EXEC PGM=&ROOTPGM.PNPR10,
```

```
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE')),
//             PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_))
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSIN    DD DSN=&&PAC7TR,DISP=(OLD,DELETE)
//PAC7TZ   DD DSN=&TY,DISP=SHR
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7CW   DD DSN=&&CONWIN,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7EV   DD DSN=&&PAC7EV,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=126,BLKSIZE=12600),
//            SPACE=(TRK,(01,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT1,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP1,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*  PRINT 'INFOPAC' MEMBER OF THE ORIGINAL STAGE     *
//*  INTO SEQUENTIAL FILE 'PAC7IP'                    *
//*-----------------------------------------------------*
//*
//TRS03   EXEC PGM=CONWRITE,MAXRC=0,
//        EXECIF=(&C1ACTION(1,8),EQ,'GENERATE')
//CONWIN   DD DSN=&&CONWIN,DISP=(OLD,DELETE)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(02,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT2,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*  COPY SEQUENTIAL 'PAC7IP' IN THE VSAM 'PAC7UP'    *
//*  DEFINED IN TSO WITH A NEW DDNAME FOR  EXIT3      *
//*-----------------------------------------------------*
//*
//TRS04   EXEC PGM=&ROOTPGM.PNPR11,COND=(00,NE,TRS02),
//        EXECIF=(&C1ACTION(1,8),EQ,'GENERATE')
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7EV   DD DSN=&&PAC7EV,DISP=(OLD,PASS)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(OLD,PASS)
//PAC7PU   DD DSN=&UP,DISP=SHR
//SYSOUT   DD DSN=&&SYSOUT3,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP3,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*           TRANSFER 'INFOPAC' ELEMENT              *
//*-----------------------------------------------------*
//*
//TRS05   EXEC PGM=C1BM3000,PARM=(PAC7BS,CXMSGSX),
//        EXECIF=(&C1ACTION(1,8),EQ,'TRANSFER')
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(OLD,PASS)
```

```
//CXMSGSX  DD SYSOUT=&OUT
//*
//*-------------------------------------------------*
//*           PRINT SYSOUTS, SYSUDUMP, ...          *
//*-------------------------------------------------*
//*
//TRS06   EXEC PGM=CONLIST,PARM=PRINT,COND=EVEN,MAXRC=0,
//        EXECIF=((&C1ACTION(1,3),NE,'ADD'),
//               (&C1ACTION(1,4),NE,'MOVE'),
//               (&C1ACTION(1,6),NE,'UPDATE'))
//C1BANNER DD DSN=&&BANNER,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//C1PRINT  DD SYSOUT=&OUT,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171,DSORG=PS)
//LIST01   DD DSN=&&SYSOUT1,DISP=(OLD,DELETE)
//LIST02   DD DSN=&&SYSDMP1,DISP=(OLD,DELETE)
//LIST03   DD DSN=&&SYSOUT2,DISP=(OLD,DELETE)
//LIST04   DD DSN=&&SYSOUT3,DISP=(OLD,DELETE)
//LIST05   DD DSN=&&SYSDMP3,DISP=(OLD,DELETE)
//*
:/ ADD NAME=PRCSDELP
//*
//*-------------------------------------------------------------------*
//*  D E L E T E    P R O C E S S O R               (PACBASE TYPE)  *
//*-------------------------------------------------------------------*
//*
//DELP    PROC ROOTPGM='BV',
//             STEPLIB='$HLQ..SBVPMBR8',
//             OUT='$OUT',
//             TY='$INDSVE..BVPTY',
//             UP='$INDSVE..BVPUP'
//*
//DEL00   EXEC PGM=BC1PDSIN,MAXRC=0
//DEL01A   DD DSN=&&SYSOUT1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//DEL01B   DD DSN=&&SYSDMP1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//DEL03A   DD DSN=&&SYSOUT2,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//DEL03B   DD DSN=&&SYSDMP2,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//*
//DEL01   EXEC PGM=&ROOTPGM.PNPR10,
//        EXECIF=(&C1COMMENT(36,5),NE,'*IBM*'),
//             PARM=(DELETE__,&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_))
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSIN    DD DUMMY
//PAC7TZ   DD DSN=&TY,DISP=SHR
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(,PASS),UNIT=SYSDA,
```

```
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7CW   DD DSN=&&CONWIN,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7EV   DD DSN=&&PAC7EV,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=126,BLKSIZE=12600),
//            SPACE=(TRK,(01,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT1,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP1,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*  PRINT 'INFOPAC' MEMBER INTO SEQUENTIAL 'PAC7IP'  *
//*-----------------------------------------------------*
//*
//DEL02   EXEC PGM=CONWRITE,MAXRC=12,
//        EXECIF=(&C1COMMENT(36,5),NE,'*IBM*')
//CONWIN   DD DSN=&&CONWIN,DISP=(OLD,DELETE)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(02,01),RLSE)
//*
//*-----------------------------------------------------*
//*  COPY SEQUENTIAL 'PAC7IP' IN VSAM 'PAC7UP'        *
//*  DEFINED UNDER TSO WITH ANOTHER DDNAME FOR EXIT3  *
//*-----------------------------------------------------*
//*
//DEL03   EXEC PGM=&ROOTPGM.PNPR11,COND=(00,NE,DEL02),
//        EXECIF=(&C1COMMENT(36,5),NE,'*IBM*')
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7EV   DD DSN=&&PAC7EV,DISP=(OLD,PASS)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(OLD,PASS)
//PAC7PU   DD DSN=&UP,DISP=SHR
//SYSOUT   DD DSN=&&SYSOUT2,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP2,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*            DELETE 'INFOPAC' ELEMENT               *
//*-----------------------------------------------------*
//*
//DEL04   EXEC PGM=C1BM3000,PARM=(PAC7BS,CXMSGSX),COND=(00,NE,DEL02),
//        EXECIF=((&C1ACTION,NE,GENERATE),
//                (&C1COMMENT(36,5),NE,'*IBM*'))
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(OLD,PASS)
//CXMSGSX  DD SYSOUT=&OUT
//*
//*-----------------------------------------------------*
//*           PRINT SYSOUTS, SYSUDUMP, ...            *
//*-----------------------------------------------------*
//*
//DEL05   EXEC PGM=CONLIST,PARM=PRINT,COND=EVEN,MAXRC=0,
//        EXECIF=(&C1COMMENT(36,5),NE,'*IBM*')
//C1BANNER DD DSN=&&BANNER,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            SPACE=(TRK,(01,01),RLSE)
//C1PRINT  DD SYSOUT=&OUT,
```

```
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171,DSORG=PS)
//LIST01   DD DSN=&&SYSOUT1,DISP=(OLD,DELETE)
//LIST02   DD DSN=&&SYSDMP1,DISP=(OLD,DELETE)
//LIST03   DD DSN=&&SYSOUT2,DISP=(OLD,DELETE)
//LIST04   DD DSN=&&SYSDMP2,DISP=(OLD,DELETE)
//*
:/ ADD NAME=PRCSMOVP
//*
//*--------------------------------------------------------------------*
//*  M O V E    P R O C E S S O R                    (PACBASE TYPE)  *
//*--------------------------------------------------------------------*
//*
//MOVP    PROC ROOTPGM='BV',
//             STEPLIB='$HLQ..SBVPMBR8',
//             OUT='$OUT',
//             TY='$INDSVE..BVPTY',
//             UP='$INDSVE..BVPUP'
//*
//MOV00   EXEC PGM=BC1PDSIN,MAXRC=0
//MOV02A   DD DSN=&&SYSOUT1,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//MOV03A   DD DSN=&&SYSOUT2,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//MOV03B   DD DSN=&&SYSDMP2,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//MOV05A   DD DSN=&&SYSOUT3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//MOV05B   DD DSN=&&SYSDMP3,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171),
//            SPACE=(TRK,(5,5),RLSE)
//*
//MOV01   EXEC PGM=&ROOTPGM.PNTRAN,
//        EXECIF=(&C1ACTION,EQ,TRANSFER),
//             PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_)M)
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7PU   DD DSN=&UP,DISP=SHR
//PAC7TR   DD DSN=&&&&C1ACTION,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=58,BLKSIZE=58),
//            SPACE=(TRK,(01,01),RLSE)
//*
//MOV02   EXEC PGM=IEBGENER,EXECIF=(&C1ACTION,EQ,MOVE)
//SYSIN    DD *
  GENERATE MAXFLDS=1
  RECORD FIELD=(058,1,,1)
//SYSUT1   DD *
,&C1SENVMNT(1,8,_),&C1SSYSTEM(1,8,_)&C1SSUBSYS(1,8,_)
&C1SELTYPE(1,8,_),&C1SELEMENT(1,10,_)&C1SSTGNUM)
/*
//SYSUT2   DD DSN=&&&&C1ACTION,DISP=(,PASS),UNIT=SYSDA,
```

```
//            DCB=(RECFM=FB,LRECL=58,BLKSIZE=58),
//            SPACE=(TRK,(01,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT1,DISP=(MOD,PASS)
//SYSPRINT DD SYSOUT=&OUT
//*
//MOV03   EXEC PGM=&ROOTPGM.PNPR10,MAXRC=0,
//             PARM=(&C1ACTION(1,8,_),&C1ENVMNT(1,8,_),
//             &C1SYSTEM(1,8,_)&C1SUBSYS(1,8,_)&C1ELTYPE(1,8,_),
//             &C1ELEMENT(1,10,_)&C1STGNUM(1,1,_)&C1CCID(1,12,_))
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//SYSIN    DD DSN=&&&&C1ACTION,DISP=(OLD,DELETE)
//PAC7TZ   DD DSN=&TY,DISP=SHR
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7CW   DD DSN=&&CONWIN,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(01,01),RLSE)
//PAC7EV   DD DSN=&&PAC7EV,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=126,BLKSIZE=12600),
//            SPACE=(TRK,(01,01),RLSE)
//SYSOUT   DD DSN=&&SYSOUT2,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP2,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*  PRINT 'INFOPAC' MEMBER FROM ORIGIN STAGE INTO    *
//*  SEQUENTIAL 'PAC7IP'                              *
//*-----------------------------------------------------*
//*
//MOV04   EXEC PGM=CONWRITE,MAXRC=0
//CONWIN   DD DSN=&&CONWIN,DISP=(OLD,DELETE)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//            SPACE=(TRK,(02,01),RLSE)
//*
//*-----------------------------------------------------*
//*  COPY SEQUENTIAL 'PAC7IP' IN VSAM 'PAC7UP'        *
//*  DEFINED UNDER TSO WITH ANOTHER DDNAME FOR EXIT3  *
//*-----------------------------------------------------*
//*
//MOV05   EXEC PGM=&ROOTPGM.PNPR11,MAXRC=0,COND=(00,NE,MOV04)
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7EV   DD DSN=&&PAC7EV,DISP=(OLD,PASS)
//PAC7IP   DD DSN=&&PAC7IP,DISP=(OLD,PASS)
//PAC7PU   DD DSN=&UP,DISP=SHR
//SYSOUT   DD DSN=&&SYSOUT3,DISP=(MOD,PASS)
//SYSUDUMP DD DSN=&&SYSDMP3,DISP=(MOD,PASS)
//*
//*-----------------------------------------------------*
//*             MOVE 'INFOPAC' ELEMENT                *
//*-----------------------------------------------------*
//*
//MOV06   EXEC PGM=C1BM3000,PARM=(PAC7BS,CXMSGSX),MAXRC=8,
//             COND=(00,NE,MOV04)
//PAC7BS   DD DSN=&&BSTIPT01,DISP=(OLD,PASS)
```

```
//CXMSGSX  DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//*
//*-------------------------------------------------*
//*           PRINT SYSOUTS, SYSUDUMP, ...          *
//*-------------------------------------------------*
//*
//MOV07   EXEC PGM=CONLIST,PARM=PRINT,COND=EVEN
//C1BANNER DD DSN=&&BANNER,DISP=(,PASS,DELETE),UNIT=SYSDA,
//            SPACE=(TRK,(01,01),RLSE)
//C1PRINT  DD SYSOUT=&OUT,
//            DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6171,DSORG=PS)
//LIST01   DD DSN=&&SYSOUT1,DISP=(OLD,DELETE)
//LIST02   DD DSN=&&SYSOUT2,DISP=(OLD,DELETE)
//LIST03   DD DSN=&&SYSDMP2,DISP=(OLD,DELETE)
//LIST04   DD DSN=&&SYSOUT3,DISP=(OLD,DELETE)
//LIST05   DD DSN=&&SYSDMP3,DISP=(OLD,DELETE)
//*
F+
//*
//
```

Initial preparation of files

D13PREP module: '$prfj.PRE' job

This preparation, which must be executed upon the first installation only, is constituted of a '$PRFJ.PRE' job which includes the following steps:

Step	Program	Comments
ET010	IDCAMS	Allocation of the VSAM file of the Endevor types of VA Pac and 'INFOPAC' elements : 'TY' Allocation of the work file of the VA Pacbase context for the elements in Endevor (update preparation) : 'UP' Allocation of the work file of the VA Pacbase + Endevor context for the elements in Endevor (update preparation) : 'UQ'
ET020	BVPNINUQ	Loading of 'UQ' RRDS file
ET030	IDCAMS	DELETE of model DSCB of 'QU' file
ET040	IDCAMS	Data-group index BLDG and initialization of 'QU' file (backup of VA Pacbase update 'UQ' file)
ET050	IEBGENER	Loading of 'QU' file
ET060	IDCAMS	Initialization of 'UP' file
ET070	IDCAMS	Initialization of 'TY' types file

Execution JCL

```
//$PRFJ.PRE JOB ($CCPT),'PREPARATION',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*
//*--------------------------------------------------------------------*
//*         JOB TO RUN ONLY FIRST TIME PACBASE IS INSTALLED            *
//*--------------------------------------------------------------------*
//*
//*****************************************************
//*       ALLOCATION TABLES AND WORK VSAM FILES         *
//*                                                     *
//* IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS *
//*****************************************************
//*
//ET010   EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=$VCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DSN=$INDSN..BVPSY(DFBVPTY),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPUP),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DFBVPUQ),DISP=SHR
//         DD DSN=$INDSN..BVPSY(DF$BASE.TS),DISP=SHR
//*
//*****************************************************
//*               LOADING 'UQ'                        *
//*****************************************************
//*
//ET020   EXEC PGM=BVPNINUQ
//STEPLIB  DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//PAC7UQ   DD DSN=$INDSVE..BVPUQ,DISP=SHR
//SYSOUT   DD SYSOUT=$OUT
//SYSUDUMP DD SYSOUT=$OUT
//*
//*****************************************************
//*       DELETE DSCB                                 *
//*****************************************************
//ET030   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DELETE ($INDUNE..DSCB.BVPQU)
//*
//*****************************************************
//*       DEFINE DSCB                                 *
//*****************************************************
//ET035   EXEC PGM=IEFBR14
//DSCBQU   DD DSN=$INDUNE..DSCB.BVPQU,DISP=(,CATLG,DELETE),
//            SPACE=(TRK,(0)),VOL=SER=$VOLUN,UNIT=$UNITUN,
//            DCB=(RECFM=FB,LRECL=187,BLKSIZE=18700)
//*
//*****************************************************
//*         BUILDING GENERATION FILE INDEX            *
//*****************************************************
//*
//ET040   EXEC PGM=IDCAMS
//*:STEPCAT   DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDUNE..DSCB.BVPQU,DISP=(,CATLG,DELETE),
//            SPACE=(TRK,0),UNIT=$UNITUN,
```

```
//            VOL=SER=$VOLUN,
//            DCB=(RECFM=FB,LRECL=187,BLKSIZE=18700)
//SYSIN    DD DSN=$INDSN..BVPSY(BLBVPQU),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//*************************************************
//*               LOADING 'QU'                    *
//*************************************************
//*
//ET050   EXEC PGM=IEBGENER
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=187,BLKSIZE=18700)
//SYSUT2   DD DSN=$INDUNE..BVPQU(+1),DISP=(,CATLG,DELETE),       ,
//            VOL=SER=$VOLUN,
//            SPACE=(TRK,(1,1),RLSE),UNIT=$UNITUN,
//            DCB=$INDUNE..DSCB.BVPQU
//*
//*************************************************
//*        INITIALIZATION OF 'UP'                 *
//*************************************************
//*
//ET060   EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=$VCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSPAF   DD DSN=$INDSVE..BVPUP,DISP=SHR
//MAXKEY   DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=$INDSN..BVPSY(REPRO999),DISP=SHR
//*
//*************************************************
//*        INITIALIZATION OF 'TY'                 *
//*************************************************
//*
//ET070   EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=$VCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSPAF   DD DSN=$INDSVE..BVPTY,DISP=SHR
//MAXKEY   DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=$INDSN..BVPSY(REPRO999),DISP=SHR
//*
//*************************************************
//*        INITIALIZATION OF 'TS'                 *
//*************************************************
//*
//ET080   EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=$VCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SYSPAF   DD DSN=$INDSVE..$BASE.TS,DISP=SHR
//MAXKEY   DD DSN=$INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=$INDSN..BVPSY(REPRO999),DISP=SHR
//*
//
```

LINK-EDIT of EXITS in an authorized library

D14EXIT module: '$prfj.EXT job

The VA Pacbase Endevor system is constituted of 2 EXITS (EXIT2 and EXIT3) which must be linked with 'EPC1UEXT' Endevor DRIVER.

The '$PRFJ.EXT' JOB includes the 2 following steps (which must be adapted to the sites) :

Step	Program	Comments
ET010	IEWL	EXIT2 link
ET020	IEWL	EXIT3 link

Note: After this job, the user must define EXITs in Endevor by adding them via the C1UEXIT macro structure lines of the 'BC1JXITS' Endevor JCL.

Execution JCL

```
//$PRFJ.EXT JOB ($CCPT),'EXIT2-EXIT3',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//*                                                                   *
//*        LINK-EDIT EXITS IN AN AUTHORIZED LINKLIST LIBRARY          *
//*                                                                   *
//*                          ----------------                         *
//*                                                                   *
//* IMPORTANT: YOU MUST DEFINED EXIT PROGRAMS TO ENDEVOR BY ADDING    *
//*            THEM TO THE àC1UEXIT MACRO IN THE ENDEVOR JCL          *
//*            'BC1JXITS'.                                            *
//*                                                                   *
//*********************************************************************
//ET010   EXEC PGM=IEWL,PARM='LIST,XREF,LET'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1024,(300,100))
//SYSLIB   DD  DSN='$HLQ..SBVPMBR8',DISP=SHR
//         DD  DSN=$LDLIB,DISP=SHR
//         DD  DSN=$CONLIB,DISP=SHR
//SYSLMOD  DD  DSN=$LDLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  *
 INCLUDE SYSLIB(EPC1UEXT)
 INCLUDE SYSLIB(C1UEXT02)
 ENTRY EPC1UEXT
 NAME C1UEXT02(R)
/*
//*
//ET020   EXEC PGM=IEWL,PARM='LIST,XREF,LET'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1024,(300,100))
//SYSLIB   DD  DSN='$HLQ..SBVPMBR8',DISP=SHR
//         DD  DSN=$LDLIB,DISP=SHR
//         DD  DSN=$CONLIB,DISP=SHR
```

```
//SYSLMOD  DD  DSN=$LDLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  *
 INCLUDE SYSLIB(EPC1UEXT)
 INCLUDE SYSLIB(C1UEXT03)
 ENTRY EPC1UEXT
 NAME C1UEXT03(R)
/*
//*
//
```

Loading of TSO / VA Pac-ENDEVOR messages

D15MSGS module: '$prfj.MSG' job

This job loads the TSO user messages of the Endevor VA Pacbase system into Endevor 'ISPMLIB' message library from the 'CIUU$MSGSX' member present in the PDS of 'SY' parameters. (See the meaning and the possible values of the '$MSGSX' parameter in the table of parameters).

This job includes the following step:

Step	Program	Comments
ET010	IEBCOPY	Loading of the 'CIUU$MSGSX' member in the Endevor 'ISPMLIB' library

Note: The member name follows the standards set by the Computer Associates company.

Execution JCL

```
//$PRFJ.MSG JOB ($CCPT),'LOAD USER MESSAGES',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*
//*********************************************************************
//*        LOADING OF PACBASE MESSAGES IN THE ENDEVOR 'ISPMLIB'       *
//*********************************************************************
//*
//ET010   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//SYSUT3   DD UNIT=$UWK,SPACE=(TRK,20)
//SYSUT4   DD UNIT=$UWK,SPACE=(TRK,20)
//IN       DD DSN=$INDSN..BVPSY,DISP=SHR
//OUT      DD DSN=$MSGLIB,DISP=SHR
//SYSIN    DD *
 C I=IN,O=OUT
 S M=CIUU$MSGSX
/*
//*
//
```

Retrievals from the 2.5 version (JJND RPTY RP25)

D162530 module: '$prfj.2530' job

This job is to be executed only for the sites which upgrade from the 2.5 version of the VA Pacbase / Endevor interface. See before "Retrieval of VisualAge Pacbase 2.5" chapter in this manual.

The job includes the following steps:

Step	Procedure	Comments
ET010	BVPJND	Retrieval of archived journal
ET020	BVPRPTY	Retrieval of 2.5 Types VSAM file into the Administration Database
ET030	BVPTYND	Loading of Types VSAM file of the installed version from data contained in the Administration Database supplied in the previous step
ET040	BVPRP25	Retrieval of the Elements managed by Endevor and contained in VA Pacbase as User Entities of the .NDENV and .NDVLM Meta Entities, into User Entities of the new model.

Execution JCL

```
//$PRFJ.253 JOB ($CCPT),'RETRIEVAL FROM 2.5 ',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*
//*********************************************************************
//*                       !!!! ATTENTION !!!!                         *
//*    THIS JOB MUST BE ONLY EXECUTED FOR A RETRIEVAL FROM 2.5        *
//*    BEFORE EXECUTING THIS JOB, IT IS ADVISABLE TO CONSULT          *
//*    THE CHAPTER "RETRIEVAL" OF THE INSTALLATION MANUAL             *
//*********************************************************************
//*
//ET010   EXEC BVPJJND,
//        JNARCH='????'              <-- ARCHIVED JOURNAL 2.5
//*
//*********************************************
//*    BEFORE EXECUTING THE FOLLOWING STEPS,   *
//* THE ADMINISTRATION DATABASE MUST BE CLOSED *
//*********************************************
//*
//ET020   EXEC BVPRPTY,
//        PAC7TY='????'              <-- TYPE FILE 'TY' 2.5
//*
//INPUT.CARTE DD *
 *ADMIN   ADMIN
//*
//ET030   EXEC BVPTYND
```

```
 *ADMIN   ADMIN
//*
//ET040   EXEC BVPRP25,
 *ADMIN   ADMIN
//        PAC7TY='????'              <-- TYPE FILE 'TY' 2.5
//*
//INPUT.CARTE DD *
          ENVIRON SYSTEM  SUBSYSTM
//*
//
```

Installation of the Development Database

Before being installed, a Development Database must be declared in the Administration Database.

If the code entered does not correspond to the provided database (BVAP) or to a database predefined in the Administration Database, then this reference is updated by UPGP in the Administration Database.

The complete declaration of a new database is done via Administrator workbench. For more information, refer to the 'Installation of Client Components' chapter and also to the 'AD workbench User's Guide'.

NOTE: The BVAP test Database, provided at installation, is already declared in the Administration Database.

The following operations must be executed every time a new Development Database is created.

CICS CSD Update

Development Database components

I00CSD module: '$prfj.I0' job

bbbb=BASE, code of the Development Database, transaction code and prefixes of the Development Database file codes selected by the user.

Files:

- bbbbAN: index file,
- bbbbAR: data file,
- bbbbAJ: transaction journal,
- bbbbAY: unsorted data

Transaction: bbbb executes the current operations in the Database (read-only, update).

Notes

The option allowing the "Dynamic Backout" (JNLSYNCWrite and RECOVery) is required for AN, AR, AJ, AY files.

The number of Strings, Indexbuffers and Databuffers allows the product to run correctly in most installations.

However, for sites containing large volumes of data and a large number of users, these parameters must be adapted.

For more details, refer to the 'Efficiency enhancement' subchapter in the Appendix of this manual.

Execution JCL

```
//$PRFJ.I0 JOB ($CCPT),'DEVPT DFHCSDUP',CLASS=$CLASSJ,
//      MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - CICSCSD                              *
//*    CICS: CSD BATCH UPDATE FOR ONE DEVELOPMENT DATABASE            *
//*********************************************************************
//DFHCSDUP EXEC PGM=DFHCSDUP
//STEPLIB  DD DSN=$CSDL,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//DFHCSD   DD DSN=$DFHCSD,DISP=SHR
//SYSIN    DD *
*********************************************************************
*              DEVELOPMENT DATABASE FILES                           *
*********************************************************************
DEFINE FILE($BASE.AN) GROUP($GROUF)
DESCRIPTION(DEVELOPMENT DATABASE INDEX)
       DSNAME($INDUV..$BASE.AN)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11) INDEXBUFFERS(10)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($BASE.AR) GROUP($GROUF)
DESCRIPTION(DEVELOPMENT DATABASE DATA)
       DSNAME($INDUV..$BASE.AR)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($BASE.AJ) GROUP($GROUF)
DESCRIPTION(DEVELOPMENT DATABASE JOURNAL)
       DSNAME($INDUV..$BASE.AJ)
```

```
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
DEFINE FILE($BASE.AY) GROUP($GROUF)
DESCRIPTION(DEVELOPMENT DATABASE EXTEND)
       DSNAME($INDUV..$BASE.AY)
       STRINGS(10)
       STATUS (ENABLED) OPENTIME(STARTUP)
       DATABUFFERS(11)
       RECORDFORMAT(F)
       ADD(YES) BROWSE(YES) DELETE(YES) READ(YES) UPDATE(YES)
       RECOVERY(BACKOUTONLY)
*********************************************************************
*              DEVELOPMENT DATABASE TRANSACTION                     *
*********************************************************************
DEFINE TRANSACTION($BASE) GROUP($GROUT)
DESCRIPTION(DEVELOPMENT $BASE TRANSACTION) PROGRAM(BVPR000)
//
```

Allocation and Loading of Database Parameters

I01SY module: '$prfj.I1' job

NOTE: This PDS contains the definitions of the Development Database files. A default size is specified for the test Database supplied at installation. As the Development Database is intended to grow in size, you are advised to modify the default value.

Step	Program	Comments
STEP1	IDCAMS	DELETE of PDS
STEP2	IEFBR14	allocation

Execution JCL

```
//$PRFJ.I1   JOB ($CCPT),'PAC I01SY',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - I01SY                                *
//*    INITIAL ALLOCATION OF THE PARAMETERS PDS OF ONE                *
//*    DEVELOPMENT DATABASE                                           *
//*       .STEP1 : DELETE                                             *
//*       .STEP2 : ALLOCATION                                         *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDUN..$BASE.SY)
```

```
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDUN..$BASE.SY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(6080,(100,,10))
//
```

I02SY module: '$prfj.I2' job

Caution

Replace :/ with ./ before submitting the job.

Step	Program	Comments
STEP1	IEBUPDTE	Loading

Execution JCL

```
===FRM TYPE=DATA
//$PRFJ.I2   JOB ($CCPT),'PAC I02SY',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - I02SY                                *
//*    LOADING OF THE PARAMETER PDS                                   *
//*     .STEP1 : LOADING PARAMETERS FOR ONE DEVELOPMENT DATABASE      *
//*    ->NOTE 1                                                       *
//*      ------                                                       *
//*      REPLACE :/ BY ./ BEFORE SUBMITTING THE JOB                   *
//*    ->NOTE 2                                                       *
//*      ------                                                       *
//*      THIS JOB CONTAINS THE SYSIN'S FOR ALLOCATING THE FILES       *
//*      THAT MAKING UP THE DEVELOPMENT DATA BASE :                   *
//*      ADAPT THE SIZES SPECIFIED IN THE 'DEFINE' STATEMENTS         *
//*      TO YOUR REQUIREMENTS.                                        *
//*********************************************************************
//*
//STEP1  EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDUN..$BASE.SY,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
:/     ADD NAME=DF$BASE.AN
 DELETE ($INDUV..$BASE.AN) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.AN)         -
                  SHR (2)     RUS   KEYS (49,0)   -
                  VOL ($VOLUV)                    -
                  CYL (20 1)                      -
```

```
                  RECSZ (59,59) )                 -
        INDEX   ( NAME ($INDUV..$BASE.AN.I)       -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDUV..$BASE.AN.D)       -
                  FSPC (10,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$BASE.AR
 DELETE ($INDUV..$BASE.AR) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.AR)         -
                  SHR (2)    RUS   NUMBERED       -
                  VOL ($VOLUV)                    -
                  CYL (20 1)                      -
                  RECSZ (144,144)  )              -
       DATA    ( NAME ($INDUV..$BASE.AR.D)        -
       CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$BASE.AY
 DELETE ($INDUV..$BASE.AY) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.AY)         -
                  SHR (2)    RUS   NUMBERED       -
                  VOL ($VOLUV)                    -
                  CYL (5 1)                       -
                  RECSZ (1018,1018)  )            -
       DATA    ( NAME ($INDUV..$BASE.AY.D)        -
       CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DF$BASE.AJ
 DELETE ($INDUV..$BASE.AJ) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.AJ)         -
                  SHR (2)    RUS   NUMBERED       -
                  VOL ($VOLUV)                    -
                  CYL (1 1)                       -
                  RECSZ (170,170)  )              -
        DATA    ( NAME ($INDUV..$BASE.AJ.D)       -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DL$BASE.JT
 DELETE ($INDUN..$BASE.JT)
:/     ADD NAME=DL$BASE.TJ
 DELETE ($INDUN..$BASE.TJ)
:/     ADD NAME=DF$BASE.FP
 DELETE ($INDUV..&USER.$BASE.FP) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..&USER.$BASE.FP)   -
                  SHR (2)     RUS   KEYS (33,0)   -
                  VOL ($VOLUV)                    -
                  CYL (1 1)                       -
                  RECSZ (33,33) )                 -
        INDEX   ( NAME ($INDUV..&USER.$BASE.FP.I) -
                  CISZ (4096) )                   -
        DATA    ( NAME ($INDUV..&USER.$BASE.FP.D) -
                  FSPC (10,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
```

```
:/     ADD NAME=DF$BASE.EM
 DELETE ($INDUV..$BASE.EM) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.EM)         -
                  SHR (2 3)   RUS   KEYS (17,0)   -
                  VOL ($VOLUV)                    -
                  CYL (1 1)                       -
                  RECSZ (90,90) )                 -
        INDEX   ( NAME ($INDUV..$BASE.EM.I)       -
                  CISZ (1024) )                   -
        DATA    ( NAME ($INDUV..$BASE.EM.D)       -
                  FSPC (10,5)                     -
        CISZ (4096) )         /*: CATALOG ($VCAT) */
:/     ADD NAME=DL$BASE.MY
 DELETE ($INDUV..$BASE.MY) NONVSAM
:/     ADD NAME=LI$BASE.AJ
 LISTCAT ENTRIES ($INDUV..$BASE.AJ) CLUSTER
PP
//
```

Initialization of Generation Data Groups

I03DEF module: '$prfj.I3' job

Definition of the Development Database files.

Step	Program	Comments
STEP1	IDCAMS	Delete/Define

Execution JCL

```
//$PRFJ.I3  JOB ($CCPT),'PAC I03DEF',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - I03DEF                               *
//*    DEFINITION OF THE VSAM FILES FOR ONE DEVELOPMENT DATABASE      *
//*     STEP1 : DELETE/DEFINE OF THE FILES                            *
//*********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD DSN=$INDUN..$BASE.SY(DF$BASE.AN),DISP=SHR
//      DD DSN=$INDUN..$BASE.SY(DF$BASE.AR),DISP=SHR
//      DD DSN=$INDUN..$BASE.SY(DF$BASE.AY),DISP=SHR
//      DD DSN=$INDUN..$BASE.SY(DF$BASE.AJ),DISP=SHR
//*
//
```

I03INI module: '$prfj.I3I' job

Allocation of the generation files (GDG) of the Development Database.

Caution

If the files are managed with SMS, delete the DD //GDGMOD lines from the IDCAMS steps before submitting the Job.

Step	Program	Comments
STEP1	IDCAMS	GDG of the PJ file
STEP2	IEBGENER	Initialization of the PJ file
STEP3	IDCAMS	GDG of the PC PD PY file
STEP4	IEBGENER	Initialization of the PC file
STEP5	IEBGENER	Initialization of the PD file
STEP6	IEBGENER	Initialization of the PY file

Execution JCL

```
//$PRFJ.I3I JOB ($CCPT),'PAC I03INI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - I03INI                               *
//*   JOB TO RUN ONLY FOR THE INITIALIZATION OF A NEW DATABASE        *
//*    . BUILDING OF INDEX DATA-GROUP FOR                             *
//*      "PC", "PJ" SAVE FILES                                        *
//*    . "PJ" FILE INITIALIZATION                                     *
//*    . LOADING OF TEST DATABASE ON "PC" FILE                        *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*    IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS            *
//*********************************************************************
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD   DD DSN=$INDUN..$BASE.PJ,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=FB,LRECL=170,BLKSIZE=27880)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$BASE.PJ) LIMIT (3) SCR)
//*
//STEP2  EXEC PGM=IEBGENER
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=170,BLKSIZE=170)
//SYSUT2   DD DSN=$INDUN..$BASE.PJ(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
```

```
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,1),
//         DCB=($DSCB,RECFM=FB,LRECL=170,BLKSIZE=27880)
//*
//STEP3  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$VCAT,DISP=SHR
//GDGMOD1  DD DSN=$INDUN..$BASE.PC,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//GDGMOD2  DD DSN=$INDUN..$BASE.PD,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//GDGMOD3  DD DSN=$INDUN..$BASE.PY,
//         DISP=(,KEEP,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,0),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$BASE.PC) LIMIT (3) SCR)
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$BASE.PD) LIMIT (3) SCR)
  DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUN..$BASE.PY) LIMIT (3) SCR)
//*
//STEP4  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=$HLQ..SBVPPF2(BVPPC),DISP=SHR
//SYSUT2   DD DSN=$INDUN..$BASE.PC(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,(350,10),RLSE),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//*
//STEP5  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=VB,LRECL=1023,BLKSIZE=27998)
//SYSUT2   DD DSN=$INDUN..$BASE.PD(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,(220,10),RLSE),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//*
```

```
//STEP6  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=VB,LRECL=1023,BLKSIZE=27998)
//SYSUT2   DD DSN=$INDUN..$BASE.PY(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=$UNITUN,
//         VOL=SER=$VOLUN,
//         SPACE=(TRK,(220,10),RLSE),
//         DCB=($DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//
```

Loading of the Test Database

I04REST module: '$prfj.I4' job

This job executes the REST procedure, using as input the backup file previously loaded on the disk in STEP4 of job '$prfj.I3I'.

To install the test Database you are provided with, run the JCL as is, without any modification.

To install a Database different from the one you are provided with, see the 'Administrator's Procedures' manual. You will find details on the REST procedure.

In this case, the Database to be restored must have been declared in the Administration Database.

If the version of your Database is previous to 3.5, see in the 'Database Retrieval' chapter, the sub-chapter dedicated to the retrieval of user parameters.

Execution JCL

```
//$PRFJ.I4  JOB ($CCPT),'PAC I04REST',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*               INSTALLATION - I04REST                              *
//*    CREATION OF THE DEVELOPMENT TEST DATABASE                      *
//*    INDICATE THE LANGUAGE CODE OF DATABASE ON Y LINE COL 10-11     *
//*********************************************************************
//TESTREST  EXEC BVPREST
 *ADMIN   ADMIN
 Y     20EN
//
```

Loading of the Development Model

I05META module: '$prfj.I5' job

This JCL must be submitted when the Development Database is installed for the first time and then each time it is reinstalled.

It is used to run the VINS procedure and to install the Development Model.

Execution JCL

```
//$PRFJ.I5  JOB ($CCPT),'PAC I05META',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*    - DEVELOPMENT DATABASE -                                       *
//*      MODEL INSTALLATION                                           *
//*      INDICATE THE FILE NECESSARY FOR REOR PROCEDURE               *
//*********************************************************************
//VINS   EXEC BVPVINS,FDIC='$HLQ..SBVPPF8(BVPMETBA)',
//       USER=ADMIN,BASE=$BASE
 *ADMIN   ADMIN   ***       VINS
/*
//*VINS.PAC7MR  DD DSN=REORFILE
```

Complement - Pac/Impact

I20GDG module: '$prfj.I20' job

This job creates the files which are necessary to the impact analysis module.

Caution

If SMS is installed on the site, delete the DD //GDGMOD lines from the steps GDGBJ and GDGBB before submitting the job.

Step	Program	Comments
INIFO1	BVPRMSYS	
INIFO2	IDCAMS	allocation of GDG for FO file
INIFO3	IEBGENER	initialization of file from scratch
INIFR1	BVPRMSYS	
INIFR2	IDCAMS	allocation of GDG for FR file
INIFR3	IEBGENER	initialization from scratch of FR file
INIFQ1	BVPRMSYS	
INIFQ2	IDCAMS	allocation of GDG for FQ file
INIFQ3	IEBGENER	initialization from scratch of FQ file
INIFH1	BVPRMSYS	
INIFH2	IDCAMS	allocation of GDG for FH file

Step	Program	Comments
INIFH3	IEBGENER	initialization from scratch of FH file

These files are described in the 'Components' chapter, 'Development Database' subchapter, 'Pac/Impact' section.

Execution JCL

```
//$PRFJ.I20 JOB ($CCPT),'PAC I05GDG',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//*             - IMPACT ANALYSIS -                                   *
//*            INSTALLATION - I05GDG                                  *
//*    BUILDING OF INDEX DATA-GROUP FOR "FH" "FO" "FQ" "FR" FILES     *
//*                                                                   *
//*  ->NOTE                                                           *
//*    ----                                                           *
//*    IF "SMS" IS INSTALLED DELETE //GDGMOD DD STATEMENTS            *
//*********************************************************************
//BVPGDG   PROC BASE=$BASE,               CODE OF VAPAC DATABASE
//         USER=TEST,            PACKAGE CODE FOR IMPACT ANALYSIS
//         INDUN='$INDUN',           INDEX OF NON-VSAM USER FILES
//*:       VSAMCAT='$VCAT',                 USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         OUT=$OUT,                            OUTPUT CLASS
//         DSCB='$DSCB',                      DSCB MODEL FILE
//         VOLS='SER=$VOLUN',               RESULTS FILE VOLUME
//         UNITS=$UNITUN,                     RESULTS FILE UNIT
//         UWK=$UWK                                 WORK UNIT
//*********************************************************************
//INIFO1 EXEC PGM=BVPRMSYS,PARM='&INDUN..&USER..&BASE'
//*-------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFO,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFO2  EXEC PGM=IDCAMS
//*---------------------
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER..&BASE.FO,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=266,BLKSIZE=26600)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFO,DISP=(OLD,DELETE)
```

```
//INIFO3  EXEC PGM=IEBGENER
//*-----------------------
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=266,BLKSIZE=266)
//SYSUT2   DD DSN=&INDUN..&USER..&BASE.FO(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=266,BLKSIZE=26600)
//*
//INIFR1 EXEC PGM=BVPRMSYS,PARM='&INDUN..&USER..&BASE'
//*-------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFR,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFR2  EXEC PGM=IDCAMS
//*----------------------
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER..&BASE.FR,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=72,BLKSIZE=21600)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFR,DISP=(OLD,DELETE)
//INIFR3  EXEC PGM=IEBGENER
//*------------------------
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=72,BLKSIZE=72)
//SYSUT2   DD DSN=&INDUN..&USER..&BASE.FR(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=72,BLKSIZE=21600)
//*
//INIFQ1 EXEC PGM=BVPRMSYS,PARM='&INDUN..&USER..&BASE'
//*-------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFQ,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFQ2  EXEC PGM=IDCAMS
//*----------------------
```

```
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER..&BASE.FQ,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=100,BLKSIZE=21600)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFQ,DISP=(OLD,DELETE)
//INIFQ3  EXEC PGM=IEBGENER
//*-----------------------
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=100,BLKSIZE=100)
//SYSUT2   DD DSN=&INDUN..&USER..&BASE.FQ(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=100,BLKSIZE=21600)
//*
//INIFH1 EXEC PGM=BVPRMSYS,PARM='&INDUN..&USER..&BASE'
//*-------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSABOUT DD SYSOUT=&OUT
//PACRIN DD DDNAME=SYSIN
//PACROU DD DSN=&&DFFH,DISP=(,PASS),UNIT=&UWK,SPACE=(TRK,1),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//INIFH2 EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VCAT,DISP=SHR
//GDGMOD   DD DSN=&INDUN..&USER..&BASE.FH,
//         DISP=(,KEEP,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,0),
//         DCB=(&DSCB,RECFM=FB,LRECL=160,BLKSIZE=24000)
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFFH,DISP=(OLD,DELETE)
//INIFH3  EXEC PGM=IEBGENER
//*-----------------------
//SYSIN    DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT1   DD DUMMY,DCB=(RECFM=FB,LRECL=160,BLKSIZE=160)
//SYSUT2   DD DSN=&INDUN..&USER..&BASE.FH(+1),
//         DISP=(,CATLG,DELETE),
//         UNIT=&UNITS,
//         VOL=&VOLS,
//         SPACE=(TRK,1),
//         DCB=(&DSCB,RECFM=FB,LRECL=160,BLKSIZE=24000)
//*
// PEND
//GDG   EXEC BVPGDG
```

```
//INIFO1.PACRIN DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME (&USERFO) LIMIT (3) SCR)
/*
//INIFR1.PACRIN DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME (&USERFR) LIMIT (3) SCR)
/*
//INIFQ1.PACRIN DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME (&USERFQ) LIMIT (3) SCR)
/*
//INIFH1.PACRIN DD *
  DEFINE GENERATIONDATAGROUP -
         (NAME (&USERFH) LIMIT (3) SCR)
/*
//
```

Complement -Modification of the transaction Output

When exiting the VA Pac 'bbbb' transaction, the system sends a CICS RETURN through the BVPR005 program called by XCTL ('bbbb' is the Database code valorized upon installation).

You may replace this program with another program, in order to return the control to a general user menu or to any other operation.

You can download the BVPR005 source from the VisualAge Pacbase Support site: http://www.ibm.com/software/awdtools/vapacbase/support.html

Chapter 4. Installation/Re-installation of Client Components

Things to Know Before Installing

- To install the VisualAge Pacbase Client components on a Windows workstation, you must have an Administrator profile.
- VA Pac Client components are installed via InstallShield for Windows Installer (ISWi).

  If Windows Installer is not installed on the workstation, it will be installed automatically.
- You also need Microsoft Windows Script, version 5.1 or higher. You can download it from the following URL:

  http://www.microsoft.com/downloads
- Both Administrator & Developer workbench and eBusiness Tools components require, for their online help, that a 4.7 or higher Netscape version or a 5.5 or higher Internet Explorer version be used.
- The installation of a Client component does not require the prior installation on the server of the VA Pac Database(s) to which it will connect.

  However, the code of each VA Pac Database you must indicate when you install some Client components will have to be strictly reused when these Databases are installed at the server level.

A number of administration actions must be carried out online in the Administrator workbench in order to make operational a VisualAge Pacbase installation or re-installation. These actions are documented in the 'AD workbench User's Guide', chapter 'Prior Administration Actions'.

Root Directory

By default, the root directory of all the VisualAge Pacbase Client components is:

```
C:\Program Files\IBM\VisualAge_Pacbase_35
```

```
If you are connected to Developer workbench and need
a simultaneous connection to the Pacbench module
of VisualAge Pacbase WorkStation, it is absolutely
necessary to install the two components in the same
root directory.
```

The directories located under this root can be modified only once, at the beginning, i.e. when the first component is installed.

NOTE: Do not use blank characters in directory names.

The other components will necessarily be installed under this root directory (whether it has been modified or not).

However, for the installation of a later version of a component, the new root name is your choice, knowing that it must be different than the current installation root (the installation script controls that).

By 'version' one must understand the version identified by the first digits, e.g. '3.5'. Versions identified by 'Vnn' actually are sub-versions.

For more information, see Subchapter 'Component Updating, Modification, or Removing'.

Installation Startup

Insert the installation CD-Rom.

The execution of setup.exe launches the graphical interface of Wizard InstallShield which will guide you through the installation.

The first panel displays the text of the Java runtime license. You agree with the terms of the license ; the next panel then asks for your identification (Name and Organization).

NOTE: All the VisualAge Pacbase Client components are installed in a shared use mode on the workstation.

Then the list of the VisualAge Pacbase Client components is displayed.

Choose the Client component you want to install.

The continuation of the installation is described in sub-chapters dedicated to each component.

Fundamentals of VA Pac Client-Server Communication

This section presents the principles of the communication between the Client components and the VisualAge Pacbase server.

The following pages contain important information essential to the choice of communication protocol and the parameterization of the associated middleware.

This information will also be useful for future installations (other Client components or new versions of already installed components).

AVAILABLE COMMUNICATION TYPES

- If the VisualAge Pacbase server runs on Windows or UNIX, the VAP Socket protocol must be used.
- If it runs on MVS/CICS, you can either use the MVS CICS Socket, MVS CICS CPI-C, or MQ-CICS Bridge protocol, depending on which protocol is in use at the server level (see 'Installation of Server Components' chapter).

  For more information on this configuration, see subchapter 'Additional Information', section 'How to configure the MVS CICS CPI-C protocol'.
- If it runs on MVS/IMS, you can use the MVS IMS Connect or the MQ-IMS Bridge protocol, depending on the protocol in use at the server level (see 'Installation of Server Components' chapter).

MIDDLEWARE

The middleware must always be installed on each developer work station. This installation starts automatically during the first installation under a given root of one of the following Client components: Administrator and Developer workbench, VisualAge Pacbase Workstation, and eBusiness Tools.

The middleware installed on each developer work station ensures direct communication between the Client component(s) and the Server.

However you can also choose a communication via a gateway.

This gateway performs a centralized and optimized management of server access.

In this context, you must also install the middleware on an intermediate server by selecting the Middleware item in the list of Client components (see corresponding subchapter).

Client components then communicate via a gateway (the VisualAge Pacbase Gateway) which runs on this intermediate server.

COMMUNICATION FILES

For the Administrator & Developer workbench and the VisualAge Pacbase WorkStation, the parameterization of the communication is made in two files: the bases.ini and vaplocat.ini. files.

The vaplocat.ini file is also used by the eBusiness Tools component.

These files are automatically created and are located in a directory named 'common'.

A reinstallation does not affect the bases.ini and vaplocat.ini files. A base_new.ini file is created only as a reference. It contains the most recent version of this file.

IMPORTANT: To add/delete VisualAge Pacbase Databases, or modify parameters related to the communication, you will have to modify these files.

For details on how data is structured within both files, see the end of this chapter (Updating communication parameters).

THE VAPLOCAT.INI FILE

- When communication is direct, the vaplocat.ini file used is located on each Developer workstation.
- When communication is via a gateway, the vaplocat.ini file used is located on the intermediate server.

In both cases this file is located in the 'common' sub-directory of the installation root directory.

The location(s) is(are) described in this file.

A location :

- identifies the protocol used to access the VisualAge Pacbase server,
- gives the physical addresses of the server for this protocol,
- defines the communication parameters required for the operation of this protocol.

THE BASES.INI FILE

The bases.ini file is found on each Developer workstation, in the 'common' sub-directory of the installation root directory.

This file contains the list of accessible VisualAge Pacbase Databases. Each Database is associated with a location.

Several Databases can be associated with the same location. The locations are defined in the other file, the vaplocat.ini file.

Administrator & Developer workbench

If IBM SDK for Java 2 is not installed on your workstation, its installation will automatically take place. For this installation, as for that of the Administrator & Developer workbench, the root being used depends on the current installation context. For complete details, refer to this chapter's first page.

The installation script then asks you to choose to install Administrator workbench or Developer workbench or both:

- Administrator workbench
- Developer workbench

  Developer workbench includes the following modules, each running independently:

  – Batch module,
  – eBusiness module, also including three of the eBusiness Tools:
    - Proxy Generator,
    - Location Editor,
    - Services Test Facility.
  – Services Modeler module (all of its functionalities being included in the eBusiness module).

Whether you install Administrator or Developer workbench or both, the CFM utility (Configuration File Manager) is always installed. CFM allows you to inhibit the display of selected browsers unused by your site's teams and/or to provide for the display of browsers specific to Meta Entities defined on site. Consult the VisualAge Support team for more information.

In the next panel, you indicate the communication mode (direct communication or gateway).

NOTE: IMPORTANT information on communication issues are given at the beginning of this chapter.

This panel does not appear if you have already installed Administrator & Developer workbench or the VA Pac WorkStation under the same root.

- If you choose the direct communication option, the middleware installation script will automatically start once the installation of the workbench is finished. It will ask you to specify a number of communication parameters.

  For complete details on this part of the installation, refer to the Middleware subchapter.
- If you choose the gateway option, enter first the IP address of the gateway here. The installation of the middleware on the Developer workstation - also necessary in this context - will then start automatically after the installation of the workbench.

Next, in this same initial context, enter the (first) Database which the Administrator and Developer workbench will access.

To do this, a window enables you to enter:

1. The name of a VA Pac Database, already installed at the server level or not. The names entered here will be displayed in the connection smartguide, thus showing users which Databases they can connect to. The name given to each Database should therefore be clear enough to be easily identified in the list of Databases proposed by the connection smartguide.
2. The Database logical code.

   Maximum length: 4 characters.

   If the Database is not installed at the server level yet, please keep this code in mind: it will have to be used again upon this installation. The codes entered here will also be displayed in the connection smartguide.

   NOTE: The Database specifically dedicated to the site administration is automatically created. Its logical code is the '****' reserved code.
3. The location name

   Maximum length: 20 characters.

   Default: Location-1

   More than one Database may be associated with one location.
4. Finally, specify the user authentication mode at connection. Refer to the Database Administrator to ensure authentication measures at the server level are imposed at the workbench level.

   You select the mode via two check boxes.

   VisualAge Pacbase signon:

   The user will have to enter his/her code and password to connect to the VA Pac Database.

   Middleware signon:

   The user will have to enter his/her code and password to connect to the host system (in the two fields displayed under 'Middleware references' in the connection smartguide).

   If only the Middleware signon box is checked, VA Pac authentication is performed by the security system.

   If both boxes are checked, the user will have to enter his/her code and password to connect to the host system and to the Database.

   In this way, you have defined the access to a first VA Pac Database. The installation script then allows you to define communication and connection to as many other Databases as necessary.

The actual installation can then start ; press the [Install] button.

NOTE: This installation is followed by the installation of the eBusiness tools (if not already installed under the same root) and -- in the initial context defined above -- of the Middleware.

START-UP FILES

The start-up files are :

- wb_admin.bat
- wb_global.bat
- wb_batch.bat
- wb_dialog.bat
- wb_eBusiness.bat
- wb_services.bat
- wb_extensibility.bat
- wb_cfm.bat

These files are to be found under the root directory of the Administrator & Developer workbench ('adworkbench').

START MENU / PROGRAMS CHOICE

Once the installation is complete, the Windows desktop includes the VisualAge Pacbase 3.5 Components section in the Start Menu/Programs choice, with the following sub-sections:

```
Administrator-Developer workbench
                                Administration
                                Batch
                                Dialog
                                eBusiness
                                Services Modeler
                                Extensibility
                                cfm
                                Global
```

Open Jade and Tidy / Publishing facility

The Publishing facility requires the installation of the Open Jade and Tidy open source utilities.

You can download them through the VA Pac Support web page at:

`http://www.ibm.com/software/awdtools/vapacbase/support.html`

Under 'Self Help' and 'Download', click the 'VisualAge Pacbase Downloads in English' link, follow the specific instructions, and open the 'Open Jade' and 'Tidy' links, accordingly.

Install both utilities in one or two directories of your choice, knowing that you will have to notify AD workbench of such directories through two dedicated Preferences.

eBusiness Tools

The eBusiness Tools are:

- Proxy Generator
- Location Editor
- Services Test Facility
- VisualAge Pacbase Web Connection (Dialog Web Revamping Generator)
- Web Application Models (WAM)

This installation allows the eBusiness Tools to be used independently of Developer workbench, without a connection to the VisualAge Pacbase server. eBusiness tools (save WAM) are installed as VisualAge for Java features and tools.

If IBM SDK for Java 2 is not installed on your workstation, its installation will automatically take place. For this installation, as for that of the eBusiness Tools, the root being used depends on the current installation context. For complete details, refer to this chapter's first page.

To start the installation, click on the [INSTALL] button.

The following lines are irrelevant as far as WAM is concerned.

The Middleware component is automatically installed following the installation of the eBusiness Tools if it does not already exist under the root of the current installation. You will then have to specify some communication parameters.

For information on this part of the installation, see the Middleware subchapter.

The middleware installed in this context allows communication between the server and the generated proxies. Communication parameters will have to be set by the developer with the Location Editor tool included in this installation.

Also, the eBusiness Tools component can run via a gateway. In this case you will also have to install the Middleware component and configure communication parameters, on the intermediate server which hosts the VisualAge Pacbase gateway.

NOTE: IMPORTANT information on communication issues is given at the beginning of this chapter.

START-UP FILES

The start-up files are :

- For the Proxy Generator:

  ```
  vapGen.exe
  ```

- For the Location Editor:

  ```
  vapLocationEditor.exe
  ```

- For the Services Test Facility:

  ```
  vapServicesTestFacility.exe
  ```

- For VisualAge Pacbase Web Connection:

  ```
  PacWebgen.exe
  ```

These files are to be found in the following sub-directory:

```
ebusinesstools\bin
```

- Unlike the other eBusiness Tools, WAM is installed in its own directory named Wam. This directory contains a zipped file and a readme file. Read this file and follow the instructions included therein.

START MENU / PROGRAMS CHOICE

Once the installation is over, the Windows desktop includes the VisualAge Pacbase 3.5 Components section in its Start Menu/Programs choice, with the following sub-sections:

```
eBusinessTools
              Location Editor
              Proxy Generator
              Services test Facility
              Dialog Web Revamping Generator
```

VisualAge Pacbase WorkStation

The root used for this installation depends on this installation's context. For complete details, refer to this chapter's first page.

The first panel invites you to select the language option of the VisualAge Pacbase WorkStation interface. The default language option is English.

In the following panel, you select the methodology to be implemented by the WorkStation.

NOTE: If you wish to install another methodology, you will have to repeat this installation process one more time.

If displayed, the 'Local Install' option must be selected.

NOTE: The 'sub-features' option is identical to the 'feature' option.

In the next panel, select the elements to install:

- One or both of the following modules:
  - Pacdesign,
  - Pacbench.
- The connection mode:
  - The connected mode where a connection to the VisualAge Pacbase Repository is systematically performed.
  - The open connection option where the user has to choose between the connected or the local mode.

In the next panel, you indicate the communication mode (direct communication or communication via a gateway).

NOTE: IMPORTANT information on communication issues are given at the beginning of this chapter.

This panel does not appear if you have already installed Administrator & Developer workbench or the VA Pac WorkStation under the same root.

- If you choose the direct communication option, the middleware installation script will automatically start once the installation of the workstation is finished. It will require the configuration of communication parameters.

  For information on this part the installation, see the subchapter Middleware.
- If you choose the gateway option, enter the IP address of the gateway here. The installation of the middleware on the Developer workstation - also necessary in this context - will then start automatically after the installation of the WorkStation.

Next, in this same initial context, indicate the first Database which the VisualAge Pacbase WorkStation will access.

To do this, a panel enables you to enter:

1. The name of a Database, already installed at the server level or not.

   The names entered here will be displayed in the connection smartguide, thus showing users which Databases they can connect to.

   The name given to each Database should therefore be clear enough to be easily identified in the list of Databases proposed by the connection smartguide.

   NOTE: If you use a customized file for the parameters, enter, after the Database name, the name of this file, framed by the '<' and '>' signs.

Complete details on these parameters are given at the end of this subchapter.

2. The Database logical code.

   Maximum length: 4 characters.

   If the Database is not installed yet at the server level, please keep this code in mind: it will have to be used again upon this installation. The codes entered here will also be displayed in the connection smartguide.

   The Database specifically dedicated to the site administration is automatically created. Its logical code is the '****' reserved code.

   NOTE: A logical code must be unique for a given location (see next item 3.)

3. The location name

   Maximum length: 20 characters.

   Default: Location-1

   More than one Database may be associated with one location.

4. Finally, specify the user authentication mode at connection. Refer to the Database Administrator to ensure authentication measures at the server level are imposed at the WorkStation level.

   You select the mode via two check boxes:

   Pacbase is secured:

   indicates that the user will have to enter his/her code and password to connect to the VisualAge Pacbase Database.

   Middleware is secured:

   indicates that the user will have to enter his/her code and password in the Middleware identification box to connect to the host system. If only the Middleware box is selected, VA Pac authentication is performed by the security system.

   If you check both boxes, the user will have to enter his/her code and password to connect to the host system and to the Database.

In this way, you have defined the access to a first VA Pac Database. The installation script then allows you to define communication and connection to as many other Databases as necessary.

The actual installation can then start ; press the [Install] button.

NOTE: This installation -- in the particular context defined below -- is automatically followed by the middleware installation.

START-UP FILE

The start-up file is :

```
pexec.exe
```

This file is to be found in the VisualAge Pacbase WorkStation root directory (SPAC\nnnl where 'nnn' identifies the WorkStation's version and 'l' its installed language).

START MENU / PROGRAMS CHOICE

Once the installation is completed, your Windows desktop includes the VisualAge Pacbase 3.5 Components section in its Menu Start/Programs choice, with the following sub-sections:

```
WorkStation
                WorkStation 3.5
                WorkStation 3.5 News
                <methodology> News
```

INSTALLATION PARAMETERS FILE

A number of the installation parameters of the VA Pac WorkStation are located in the Pacbase.dat file.

The WorkStation installation procedure automatically creates this file in the \SPAC\NNNL directory where 'NNN' indicates the version and 'L' the language code of the version installed.

The Pacbase.dat file, which necessarily conforms to the most recent installation, is therefore used by default when the WorkStation is started up.

However you can create one or more parameter files. This can be useful if more than one methodology is installed on a workstation, which is rather rare. It will then be easier to change the methodology when reconnecting.

The choice of file name is open but must conform to DOS file standard. The .dat extension is recommended.

These DOS files should resemble the Pacbase.dat file and should be stored in the same directory as this file.

When the VA Pac WorkStation is reinstalled, the *.dat files you created will not be deleted.

DESCRIPTION OF THE PARAMETERS FILE

Each of the lines in this file has the following structure:

- a three-digit identifier in positions 1 to 3

- the parameter label, whose position is unfixed
- the parameter value, between brackets ([and]), whose position is also unfixed

The following is an example of a PACBASE.DAT file:

```
001 Station Version              [350F]
002 Server                       [PACBASE]
003 Communication Manager        [MWCOM]
004 Communication Parameters     [MWCOM]
005 System                       [WINDOWS]
006 Method                       [MER]
007 EXE disk                     [C]
008 EXE disk(default)            [C]
009 System Data Disk             [C]
010 User Data Disk               [C]
011 Connection execution mode    [E]
```

The Pacbase.dat file should not be destroyed.

The possible values for the Methodology parameter are:

Parameter value	Methodology name
MER	MERISE
DON	YSM
ADM	SSADM (in English only)
OMT	OMT

WARNING: The parameters 001 to 005 and 011 cannot be modified.

Web Application Models (WAM)

The root used for an installation depends on the context of that installation. For more information, see the first page of this chapter.

To start the WAM installation, press the [Install] button.

WAM is installed in its own directory named Wam. This directory contains a zipped file and a readme file. Read this file and follow the instructions included therein.

Middleware

The specific installation of the Middleware component on a dedicated machine (intermediate server) is necessary only when a communication via a gateway is used.

In fact, the Middleware component is automatically installed, immediately after the first installation (under a given root) of one of the other Client components.

The root used for an installation depends on the context of that installation. For more information, see the first page of this chapter.

To use the Administrator and Developer workbench or the VA Pac WorkStation, the location parameters of your VisualAge Pacbase Databases must always be specified.

NOTE: IMPORTANT information related to the communication is given in the beginning of this chapter.

- If communication is provided via the VisualAge Pacbase Gateway, installation of the Middleware on this intermediate server requires the definition of the location necessary for the first VA Pac Database.

NOTE: In the New location field, enter a name for each location.

WARNING: If there is more than one location to define, either for the same Database or to manage more than one Database, you must define these extra locations directly in the vaplocat.ini file.

For more information on updating this file, see subchapter 'Complementary Information', section 'Updating Communication Parameters'.

- If communication is direct, the locations are automatically displayed, as they have been predefined in the first phase of the Administrator & Developer workbench or the VA Pac WorkStation installation.

Next, whatever the Client component concerned, you have to specify a certain number of different parameters, depending on the protocol used.

LIST OF PARAMETERS

- VAP SOCKET
  - IP address: IP address and port used by the VA Pac server.
    Do not enter left aligned '0', source of error with some configurations.
- MVS CICS SOCKET
  - IP address: IP address and port used by the VA Pac server.

Do not enter left aligned '0', source of error with some configurations.

- Transaction code: Code of the CICS transaction of the VA Pac Communication Monitor.
- Code Page: Value identifying the coding of characters used by the VA Pac server.

  1140 (US EBCDIC) or 1146 (UK EBCDIC)

- MVS CICS CPI-C
  - Destination-id entry: BVPSCPI (default value).

    If you modify this value, it must be the same as the value entered in the Symbolic destination name, a parameter included in the configuration of this communication protocol.
  - Code page: Value identifying the coding of characters used by the VA Pac server.

    1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MQ-CICS Bridge and MQ-IMS Bridge
  - LocCcsid: Character set code of the machine in local (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.

    Maximum: 9 characters.
  - Queue Manager name: Name of the local Queue Manager of the client part.

    If there is an in-between MQSeries Server on NT, this parameter must identify the in-between NT Queue Manager and not the Queue Manager.

    Maximum: 4 characters.
  - Request Queue name:

    Maximum: 48 characters.
  - Reply Queue name:

    Maximum: 48 characters.
  - Transaction code:

    Maximum of 4 characters for CICS.

    Maximum of 8 characters for IMS.
- MVS IMS Connect
  - IP address: IP address and port used by the VA Pacbase server.
  - Transaction code: IMS transaction code of the VA Pacbase Communications Monitor.
  - Code page: Value identifying the coding of characters used by the VA Pacbase server.

    1140 (US EBCDIC) or 1146 (UK EBCDIC)
  - Data Store:

Name of the link to IMS defined in IMS Connect (IMS Data Store ID).

– RACF group: Name of the RACF group for IMS Connect.

You can now start the installation; press the [Install] button.

Additional Information

How to Configure the MVS CICS CPI-C Protocol

The following text gives you details on the configuration specific to the use of VisualAge Pacbase Client components.

The prerequisites are:

- IBM Personal Communications 4.2
- Microsoft SNA Server 3.0A

  The Service Pack 2 must be installed for APPC corrections.

The protocol (CPI-C/APPC LU6.2) must be configured as follows:

- In the ″Define CPI-C Side Information″ window, type BVPSCPI in the ″Symbolic destination name″ field.

  This code identifies the monitor which sets up the connection between the Client components and the VisualAge Pacbase server programs which use APPC.

NOTE: It is possible to type a value other than BVPSCPI, provided the Symbolic destination name value is the same as the MONITOR parameter value in the vaplocat.ini file.

Example of a location in vaplocat.ini:

```
<MVS-DB2-CPIC>
COMM_TYPE=CPIC
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1140 (US EBCDIC) or 1146 (UK EBCDIC)
```

- In the ″TP name″ field, type the CICS transaction code which is associated with the BVPSCPI communication monitor.

Editing Communication Parameters

The bases.ini File

You will need to update the bases.ini file to add or delete a Database, or to modify communication parameters.

By default this file's access path is:

```
C:\Program Files\IBM\VisualAge_Pacbase_35\Common\
```

This file's format meets the standards of Windows .ini files.

Each section in the bases.ini file defines a configuration allowing access to one VisualAge Pacbase Database. Each section's name must be framed by brackets [Section Name].

The name of each section will be presented to the user in the connection smartguide. In the displayed list of VA Pac Databases, the user picks the Database he/she wants to connect to. This is why section names need be very explicit. All the more so since you can manage several communication options for one VA Pac Database. To do so, define as many configurations/sections as needed for one Database, clearly differentiated from one another by their name.

NOTE: With the VA Pac WorkStation, you may use a customized parameters file. To do so, enter -- after the VA Pac Database name -- this file's name framed by the '<' and '>' signs. Complete details on these parameters are given above, at the end of the 'VisualAge Pacbase WorkStation' subchapter.

DESCRIPTION OF A SECTION'S CONTENTS

The parameters in each section are listed below. There one parameter per line:

- baseCode = code of the VisualAge Pacbase Database (required)

  Maximum length: 4 characters

  NOTE: Concerning the VA Pac WorkStation, this code must be unique in the bases.ini file for a given Location.
- signOn = indicator for the control of the user signon. This indicator is required and takes one of the three following values:
  - VAPac: indicates that the user will have to give his/her code and password only when he/she connects to the VisualAge Pacbase Database.
  - Middleware: indicates that the user will have to give his/her code and password only when he/she connects to the host. The connection to the VA Pac Database will be controlled by RACF (or equivalent).
  - VAPac Middleware: indicates that the user will have to give his/her code and password when he/she connects to the host and to the Database (default option).
- communicationAdapter = indicates the communication mode in use.
  - DIRECT: local middleware
  - GATEWAY: remote middleware (via the VisualAge Pacbase gateway)

The following parameters vary according to the chosen option.

- connectionCleaningInterval = period of time (in millisecond) between two startups of connections cleaning procedures.

  Parameter needed for the VisualAge Pacbase Workstation only.

  Default value for Windows NT/2000/XP = 60000.

  For Windows 95/98, this parameter must be set to -1, and cannot be modified, to avoid problems of multi-threading, not supported by these platforms.

PARAMETERS FOR DIRECT ADAPTER (LOCAL MIDDLEWARE)

- locationsFile = indicates the path and name of the file which contains the locations definitions.

  Default: ..\common\vaplocat.ini

  CAUTION: The default value of this parameter should not be modified.
- location = location name for the Database

  Maximum length: 20 characters.

  Default: Location-1

  More than one Database can point to the same location.

  REMINDER: A location identifies the communication protocol used to access the VisualAge Pacbase server and the physical address of this server for this protocol.
- traceFile = path and name of the file which will receive the trace of the middleware execution.

  By default this file is automatically created (with timestamp) in the VapTrace sub-directory.
- traceLevel = trace level of the middleware execution. Its possible values are:
  - 0 : no trace
  - 1 : trace with errors (default)
  - 2 : standard trace, not detailed
  - 3 : trace for information
  - 4 and + : trace for debug
- codePageFile = path and name of the file which contains the conversion table of the code pages.

  Default: ..\middleware\CharConv.txt

PARAMETERS FOR GATEWAY ADAPTER (REMOTE MIDDLEWARE)

- host = name or IP address of the host where the VisualAge Pacbase gateway is installed.

Do not enter left aligned '0', source of error with some configurations.

Default: 127.0.0.1 for a local host

- port = value of the IP port where the gateway receives the client requests.

  Default: 5647
- location = location name for the Database

  Maximum length: 20 characters.

  Default : Location-1

  More than one Database can point to the same location.

  REMINDER: A location identifies the communication protocol used to access the VisualAge Pacbase server and the physical address of this server for this protocol.
- The traceFile and traceLevel parameters can be used to start the GATEWAY. For more details, refer to the Middleware User's Guide.

NOTE: All the parameters which may be present in the bases.ini file are not explained here. In fact, a number of these parameters allow finer middleware settings, particularly used by proxies (generated by the eBusiness Tools). These parameters are used separately from the bases.ini file and are documented in the Proxy Programming Interface manual.

The vaplocat.ini File

You will have to update the vaplocat.ini file to add or delete a Database, or possibly modify other parameters described below.

By default, the path to this file is:

```
C:\Program Files\IBM\VisualAge_Pacbase_35\Common\
```

To add a VisualAge Pacbase Database, create a line on which you enter the location name between '<' and '>'.

The maximum length of this name is 20 characters.

According to the protocol selected, you will have to choose different parameters (one per line):

- VAP SOCKET

  ```
  <LocationName>
  COMM_TYPE=SOCKET
  MONITOR=BVPSCPI
  MESSAGE_LENGTH=31744
  IXO_TIMEOUT=30
  IXO_ADDRESS=127.0.0.1 3676
  ```
- MVS CICS SOCKET

```
<LocationName>
COMM_TYPE=TCPMVS
MONITOR=BVPSSOC
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1140 (US) or 1146 (UK)
IXO_ADDRESS=127.0.0.1 3676
IXO_TRANSID=V303
```

- MVS CICS CPI-C

```
<LocationName>
COMM_TYPE=CPIC
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1140 (US) or 1146 (UK)
```

- MQ-CICS Bridge

```
<LocationName>
MESSAGE_LENGTH=31744
MONITOR=BVPSCPI
COMM_TYPE=MQCICS
IXO_LOCALCCSID=819
IXO_QUEUEMANAGER=EECC
IXO_REQUESTQUEUE=VAP.CICS_BRIDGE.VAPCS.REQUESTER
IXO_REPLYQUEUE=VAP.CICS_BRIDGE.VAPCS.REPLY
IXO_DYNAMICREPLYQUEUE=VAP.CICS_BRIDGE.VAPCS.DYNAMICREPLY
IXO_TRANSID=WK51
IXO_REQUESTEXPIRY=120
IXO_TIMEOUT=35
IXO_HEADERVERSION=1
```

- IMS Connect

```
<LocationName>
COMM_TYPE=TCPIMS
MONITOR=XXXXXXXX
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1140 (US) or 1146 (UK)
IXO_ADDRESS=127.0.0.1 3676
IXO_TRANSID=WK35
IXO_DATASTORE=IMSC
IXO_RACFGROUP=FR42
```

- MQ-IMS Bridge

```
<LocationName>
MESSAGE_LENGTH=31744
MONITOR=XXX
COMM_TYPE=MQIMS
IXO_LOCALCCSID=819
IXO_QUEUEMANAGER=EECC
IXO_REQUESTQUEUE=VAP.IMS_BRIDGE.VAPCS.REQUESTER
IXO_REPLYQUEUE=VAP.IMS_BRIDGE.VAPCS.REPLY
```

```
IXO_DYNAMICREPLYQUEUE=VAP.IMS_BRIDGE.VAPCS.DYNAMICREPLY
IXO_TRANSID=WK51
IXO_REQUESTEXPIRY=120
IXO_TIMEOUT=35
```

DETAILS ON THE PARAMETERS

The following list is organized according to the alphabetical order of the parameters.

- COMM_TYPE:

  This parameter identifies the communication protocol in use.

  The possible values are:

  – SOCKET: VA Pac Server under Windows or UNIX, with the use of TCP/IP.

  – TCPMVS: VA Pac Server under MVS/CICS with the use of a TCP/IP listener.

    .CPIC:: VA Pac Server under MVS/CICS, with the use of the CPI-C protocol.

  – MQCICS: VA Pac Server under MVS/CICS, with the use of the MQ-CICS-BRIDGE protocol.

  – MQMCICS: Same as MQCICS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.

  – TCPIMS: VA Pac Server under MVS/IMS, with the use of the IMS Connect protocol.

  – MQIMS: VA Pac Server under MVS/IMS, with the use of the MQ-IMS-BRIDGE protocol.

  – MQMIMS: Same as MQIMS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.

- IXO_ADDRESS: IP address and port used by the VA Pac Server.

  Do not enter left aligned '0', source of error with some configurations.

  The port number must correspond to the one indicated at the host machine configuration for OS390 CICS and IMS/VS.

  The same for Windows or UNIX, but for these platforms, a number specifically identifies a VA Pac Database. For more details, refer to the 'Installation of Server Environment' chapter, 'Repository Installation' subchapter, 'Development Databases' section, in the respective Installation Guides.

- IXO_DATASTORE: Name of link to IMS defined in IMS Connect (IMS DataStore ID).

- IXO_DYNAMICREPLYQUEUE

MQ-CICS Bridge and MQ-IMS Bridge.

When this parameter is set, the reply Queue is dynamically created by MQSeries to pass the response messages (consult the MQSeries documentation to use this type of Queue). This parameter must contain the dynamic name of the Queue (Dynamic Queue name) and the IXO_REPLYQUEUE parameter must contain the 'Queue Model' name (48 characters maximum).

- IXO_HEADERVERSION: Version of the MQCIH structure:

  MQ-CICS Bridge.

  1 (default) or 2, depending on the used CICS Bridge interface. (1 character maximum).
- IXO_LOCALCCSID: code of the character set of the machine locally

  MQ-CICS Bridge and MQ-IMS Bridge.

  This code (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.

  (9 characters maximum).
- IXO_QUEUEMANAGER: Local Queue Manager name of the client part.

  MQ-CICS Bridge and MQ-IMS Bridge.

  (4 characters maximum).

  If there is an itermediate MQSeries Server on NT, this parameter identifies the intermediate NT Queue Manager and not the Queue Manager.
- IXO_RACFGROUP: Name of RACF group for IMS Connect.
- IXO_REPLYQUEUE: Name of reply messages Queue.

  MQ-CICS Bridge and MQ-IMS Bridge.

  (48 characters maximum).
- IXO_REQUESTEXPIRY: Expiration delay of the request message, in seconds.

  MQ-CICS Bridge and MQ-IMS Bridge.

  (9 characters maximum).
- IXO_REQUESTQUEUE: Name of request messages Queue.

  MQ-CICS Bridge and MQ-IMS Bridge.

  (48 characters maximum).
- IXO_TIMEOUT: Maximum time required for a workstation to receive an answer from the server before indicating any communication error.

  This parameter is indicated in seconds. Its default value is 30.
- IXO_TRANSID: Transaction code.

  (4 characters maximum for CICS and 8 characters maximum for IMS).

  This IMS transaction will have to be declared on IMS by a GEN INPUT IMS as follows:

```
APPLCTN  PSB=BVPSSOC
TRANSACT CODE=WK35,SEGSIZE=32000,MODE=SNGL,SEGNO=00050,
         PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
         MSGTYPE=(MULTSEG,RESPONSE,$CLS)
```

- HOST_ENCODING: Identifies the encoding of the characters used by the VisualAge Pacbase server.

  1140 (US EBCDIC) or 1146 (UK EBCDIC)

  Not used with MQ-CICS Bridge and MQ-IMS Bridge.
- MESSAGE_LENGTH The value of this parameter MUST be 31744.
- MONITOR: Communication monitor code for VisualAge Pacbase, which is BVPSCPI, or BVPSSOC for MVS CICS SOCKET.

  Not used with IMS Connect and MQ-IMS Bridge.

  NOTE: For MVS CICS CPI-C, you can however enter a value other than BVPSCPI, bearing in mind that the value of the MONITOR parameter must be in all cases the same as the one set in the Symbolic destination name, a parameter included in the communication protocol configuration.

All the parameters of the vaplocat.ini file are not explained here. In fact, certain parameters allow finer middleware settings, particularly used by proxies (generated by eBusiness Tools). In this context, these parameters are edited with the Location Editor tool and are therefore documented in its online help. You can also consult the documentation of these parameters in the Middleware use's Guide.

Component Updating, Modification, or Removing

The VA Pac Client Components Installation CD-Rom also allows you to:

- 'Update' a component which is already installed.
- 'Modify' i.e. add a new sub-component to a component which is already installed.

  This option is available only with:
  - Administrator & Developer workbench:

    To add one of both workbenches as the other is already installed,
  - VisualAge Pacbase WorkStation:

    To add a Methodology.
- 'Remove' a component which is already installed.

  You can also use the Windows service 'Add/Remove' Programs in the Configuration Panel.

Chapter 5. Tests

List of Utilities

The summary table below lists the management utilities of the Administration and Development Databases.

JCL	Description
JCLARAD	Archiving of the Administration Database journal
JCLSVAD	Saving of the Administration Database
JCLROAD	Reorganization of the Administration Database
JCLRSAD	Restoration of the Administration Database
JCLPAGX	Extraction of Administration Database
JCLUPGP	Update of PAF batch format of Administration Database
JCLARCH	Storage of the Development Database journal
JCLSAVE	Saving of the Development Database
JCLMLIB	Library management
JCLREOR	Reorganization of the Development Database
JCLREST	Restoration of the Development Database
JCLUPDT	Batch update of the Development Database
JCLGPRT	Generation print
JCLEXLI	Library extraction
JCLEXPJ	Journal extraction
JCLEXTR	Entities extraction
JCLUXSR	sub-networks extraction

Installation Tests

The VA Pac Installation tests include the following operations:

- Generation-print, online and batch update tests,
- Administration procedures tests,
- Development procedures tests,
- Extraction utility tests.

Generation-Print, Online and Batch Update Tests

These tests consist of the following steps:

- On-line use tests:
  - Opening the test Database files in on-line mode.
  - Testing screen branching.
  - Executing some updates.
- Batch updating tests:
  - Executing the 'JCLUPDT' JCL (UPDT procedure).
  - The Database files must be closed to on-line use.
- Test on generation and print of programs:
  - Executing the 'JCLGPRT' JCL (GPRT procedure).

Administration Database Procedures Tests

You must first consult and perform a number of updates with the Administrator workbench.

You can then carry out the procedure tests, knowing that the Administration Database files must be closed to on-line access.

These tests include the following steps, to be executed in this order:

- Archiving of the journal created during the use tests: execute the 'JCLARAD' JCL, which outputs a PK(1) file.
- Backup of the Administration Database: execute the 'JCLSVAD' JCL, which outputs a PE(1) file; as the evaluation key does not allow to execute this procedure, you must enter the previous provided access key.
- Reorganization of the sequential backup, PE(1), of the Administration Database: execute the 'JCLROAD' JCL, which outputs a PE(2) file.
- Restoration of the Administration Database using the PK(1) archived transaction file and the PE(2) Database backup file: execute the 'JCLRSAD' JCL.
- Extraction of Administration Database data: execute the 'JCLPAGX' JCL and save the GY file of extracted data.
- PAF update of the Administration Database: execute the 'JCLUPGP' JCL with the extracted file (result of the 'JCLPAGX' JCL execution).

Development Database Procedures Tests

You must first consult and perform a number of updates with the Developer workbench.

You can then carry out the procedure tests, knowing that the Development Database files must be closed to on-line access.

These tests include the following steps, to be executed in the following order:

- Archiving of the journal created during the use tests: execute the 'JCLARCH' JCL, which outputs a PJ(1) file.
- Direct backup of the Development Database: execute the 'JCLSAVE' JCL, which outputs a PC(1) file; as the evaluation key does not allow to execute this procedure, you must enter the previous provided access key.
- Library manager: add/delete a library in the Development Database: execute the 'JCLMLIB' JCL, which outputs a PC(2) file.
- Reorganization of the sequential backup, PC(2), of the Development Database: execute the 'JCLREOR' JCL, which outputs a PC(3) file.
- Restoration of the Development Database using the PJ(1) archived transaction file and the PC(3) Database backup file: execute the 'JCLREST' JCL.

The Development Database files must be closed to on-line use while these tests are being performed.

It is advised to briefly test on-line operations again, after restoring and re-opening the Development Database files to make sure that the application runs properly.

Extraction-Utility Tests

The purpose of these tests is to execute the Database extraction procedures.

These tests include the following steps, to be executed in the following order:

- Extraction of a library as transactions: execute the 'JCLEXLI' JCL.
- Extraction of entities from a library: execute the 'JCLEXTR' JCL.
- Extraction of selected transactions and/or lists of transactions from the archived journal (PJ): execute the 'JCLEXPJ' JCL.
- Extraction of sub-network: execute the 'JCLUXSR' JCL.

To run these tests, the development files can be open in on-line mode.

Each of these jobs can be followed by an UPDT or UPDP procedure to check the validity of these extracted transactions.

Chapter 6. Re-installation of Server

A reinstallation of the system environment of the VisualAge Pacbase server is necessary in the case of corrections or enhancements on the version installed.

To install this version, download the cartridge in the dedicated PDS by using SMP/E and execute the JCLs supplied if necessary.

This version is identified by a number and includes:

- An installation cartridge (or tape),
- The "Program Directory for VA Pacbase" specific to SMP/E,
- The list of corrected anomalies,
- Instructions -- possibly included -- to complete this chapter.

Generally, only system files and program libraries are impacted by this version.

In any case, load-modules are updated by SMP/E. They are copied in hlq.SBVPMBR8 and hlq.SBVPMTR8 PDS.

Remark in case of a previous SMP/E installation

The SMP/E reinstallation implies that the SMP/E context is empty for the product: either the product has never been installed with SMP/E, or the files related to SMP/E have been reinitialized for the previous version, or you want to create a new SMP/E environment for this reinstallation.

Otherwise, if you want to re-use the SMP/E files of the previous version, you can execute, before step 1, the SMP/E utility described in the Appendix at the end of this manual.

Two cases are possible:

Case 1: Standard re-installation

This operation consists in executing the jobs contained in the following JCL modules:

1. D04MBR : job $prfj.D4B renommage load-modules exit-users batch,
2. D06SKEL: loading skeleton files,
3. D07AE0 : loading error messages,

4. D08XMET: installation of the Administration Model, (see the description in the appendix at the end of the manual).
5. I05META: installation of Development Model extension. This job must be run for each re-installed base (see the description in the appendix at the end of the manual).

Notes

If the execution report of the D08XMET JCL indicates that a re-organization is requested, you will have to run the following JCLs:

1. JCLSVAD: backup of the Administration Database,
2. JCLROAD: re-organization of the Administration Database,
3. JCLARAD: archiving of the Administration Database journal,
4. JCLRSAD: restoration of the Administration Database.

If the execution report of the I05META JCL indicates that a re-organization is requested, you will have to run the following JCLs:

1. JCLSAVE: backup of the Development Database,
2. JCLREOR: re-organization of the Development Database,
3. JCLARCH: archiving the Development Database journal,
4. JCLREST: restoration of the Development Database.

Case 2: Non-standard re-installation

It consists in executing specific operations detailed in the instructions included in the shipment.

Here are some general rules you must follow before executing the jobs previously described in the standard re-installation.

- If a JCL, a procedure or a SYSIN has been modified, you must rerun the hlq.SBVPINST(BVPPINIT) JCL and split the result file to re-install the JCLs as described in the chapter 3, 'Parameterization' subchapter.
- If a system or an Administration Database SYSIN has been modified, you must execute the D02CPAR JCL.
- If there is a new skeleton or system file, you must modify the D03DEFIN JCL and keep the DELETE/DEFINE of the new files only, then execute this JCL. And if needed, execute the D03INI JCL on the concerned files to re-initialize them.
- If there is a new procedure, you must run the D05PROC JCL.
- If there is a new SYSIN of the Development Database, execute the I02SY JCL.

Note:

The split of the BVPPINIT JCL result file, as well as the D02CPAR, D05PROC and I02SY JCLs respectively override all the JCLs, the system and Administration Database SYSINs, the procedures, and the Development Database SYSINs.

It is recommended to previously save the customized elements.

Chapter 7. Retrieval - Exchanges between 2.n & 3.n Databases

Retrieval of VisualAge Pacbase 2.0 and 2.5

Foreword

The installation of the release can be completed by the execution of utilities procedures.

These procedures are dedicated to prepare the retrieval in the new release.

There are two types of procedures:

Procedures to be executed on the 2.5 Database.

See the 'Help to retrieve 2.5 utilities' manual for a detailed documentation.

These procedures are :

- UTAG : AG file purge,
- UTFG : PIA stamp,
- UTSD : association of keyword to a data structure type.

Procedures to be executed on the new Database.

See the 'Appendix' at the end of this manual for a detailed documentation.

These procedures are :

- UTU1 : 'UNS' lines extraction,
- UTU2 : 'UNS' lines update.

Operations to be Performed

The installation of this version requires, in the one hand, the retrieval of the AG (generation-print commands file), AE AP (user parameters files) and AB AC (PEI files) files in the new Administration Database, and on the other hand, the retrieval of the old Development Database.

Sequence of operations

It consists of seven steps:

1) Backup of all the old files required. You must execute the following procedures in the old version.

- SAVE: backup of the Development Database (PC),
- PARM: backup of the user parameters (PE),
- SVAG: backup of the generation-print commands (PG)
- SVPE: backup of the PEI environment (PP).

2) Installation of the Administration Database

To install the Administration Database, execute the installation process up to the D07AE0 JCL.

This step creates the GN, GR, GY, GJ and GU files.

You must execute the following JCLs:

- Creation of the Administration Database,
  - D08INGU: creation and initialization of GU user codes file,
  - D08INAD: creation and initialization of the Database backup file (PE file), and journal backup file (PK file),
  - D08RSAD: initialization of the Administration Database with installation data,
  - D08TINQJ: initialization of the QJ archive file
  - D08XMET: installation of the Administration Model (see the Appendix, at the end of the manual),
- Implementation of the access key from Administrator workbench by executing the following operations:
  - Input of access key,
  - targets definition,
  - key activation.
- Re-organization of the Administration Database if it is mentioned in the execution report of the preceding job (D08XMET).
  - JCLSVAD: backup of the Administration Database,
  - JCLROAD: re-organization of the Administration Database,
  - JCLARAD: initialization of the Administration Database journal file,
  - JCLRSAD: restoration of the Administration Database,
- Retrieval of the old Database data,
  - RPE250: retrieval of user parameters from the PE file which was generated during step 1,
  - JCLUTMP: retrieval of 2.0 user passwords (optional),
  - JCLSVAD: backup of the Administration Database.

3) Retrieval of a Development Database.

To perform this step, the installation process of the Development Database(s) must be run up to the I03INI JCL.

It consists in executing the following JCLs:

- RPC250: retrieval of the old Development Database from the backup of the old Database which was created during step 1, and re-organization of the new Database.
- JCLARCH: archiving of the Development Database,
- I04REST: restoration of the new Development Database from the backup obtained previously; do not forget to specify the Development Database code in the user input of the procedure,
- I05META: installation of the new Database Development Model (see the description in the appendix at the end of this manual and create the MR file to take its transactions into account in the re-organization),

The execution of the following procedures, even not requested in the I05META report, is advised for a better optimization.

- JCLSAVE: backup of the new Development Database,
- JCLREOR: re-organization of new Development Database,
- JCLREST: restoration of the Development Database from the backup file resulting from the preceding re-organization procedure.

Steps 4, 5, 6 and 7 are optional.

4) Retrieval of generation-print commands.

It involves executing the following JCL:

- RPG200: if retrieval of 2.0 PG file,
- RPG250: if retrieval of 2.5 PG file.

5) Retrieval of Pac/Transfer parameters (UV).

It involves in executing the following JCL's:

- RUV250: retrieval of the UV file data.

6) Retrieval of PEI files.

It involves executing the following JCL:

- RPP250: retrieval of the PP file data.

7) Retrieval of PJ journal

It involves executing the RPJ250 JCL.

Retrieval of User Parameters (PE25)

PE25 - Introduction

Principle

This procedure (PE25) retrieves the PE file resulting from the user parameters backup executed by the PARM procedure, to update the Administration Database.

Execution conditions

The Administration Database files must be closed to on-line use.

Printed output

This procedure prints a report which indicates the errors encountered.

Result

This procedure integrates the 2.0 or 2.5 user parameters in the Administration Database.

Note

If the Database to be retrieved is under RACF control, you cannot connect via the ADMIN code provided upon installation ; you must use an Administrator code of this Database.

PE25 - Input / Processing / Results

A '*' line in which you indicate a user code and password.

An 'A' line (optional) in which you indicate the Administrator's code and name. This line is necessary only if a security system (RACF) is used.

The 'A' line has the following structure:

Position	Length	Value	Meaning
2	1	'A'	Line code
3	8	bbbbbbbb	Administrator's code
11	36		Administrator's name

A 'B' line by Database. You indicate in this line the characteristics of the Development Databases that will be managed in the new Administration Database. You must specify:

- the Database code: it is the logical code, which will be indicated upon the Database restoration.
- the Database name
- the transaction code: it is used to connect to the Database in character mode. The $BASE installation parameter is also used to code the file names.

If the Database code or name is not specified, an error message is sent and the procedure cannot be run.

The 'B' line has the following structure:

Position	Length	Value	Meaning
2	1	'B'	Line code
3	4	bbbb	Logical Database name
7	36		Database name
43	4	cccc	Transaction code

A 'C' line (optional) in which you indicate the code of the Database linked to the Optional Command Lines Set. If the command is common to all the Databases, do not enter this line and the Database code is '****'.

The 'C' line has the following structure:

Position	Length	Value	Meaning
2	1	'C'	Line code
3	4	bbbb	Database code

PE25 - Description of Steps

Input recognition: PTU001

Processing of user parameters (PE): PTU920

Code	Physical name	Type	Label
PAC7EN	&OLDPE	Input	User parameters, old version
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7MB	&&PE25MB	Input	User input
PACGGR	&INDSV..BVPGR	Input	Administration Database data

Code	Physical name	Type	Label
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGY	Output	User parameter transactions (length=310)
PAC7ET		Report	Error report

Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Input	Administration Database data
PAC7AN	&INDSV..BVPGN	Input	Administration Database index
PAC7AE	&INDSV..BVPAE	Input	Error labels
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Output	Administration Database Data file
PAC7AN	&INDSV..BVPGN	Output	Administration Database Index file
PAC7AY	&INDSV..BVPGY	Output	Administration Database extension
PAC7AJ	&INDSV..BVPGJ	Output	Administration Database journal
PAC7AE	&INDSV..BVPAE	Input	Error messages

Code	Physical name	Type	Label
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of transactions specific to a user is preceded by a banner with this user's code.

Return codes:

- 0 : OK without error
- 2 : warning error
- 4 : fatal error

PE25 - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*                 RETRIEVAL OF PE FILE
//*
//* -------------------------------------------------------
//*
//BVPPE25  PROC OUT=$OUT,                              OUTPUT CLASS
//         INDSV='$INDSV',                INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',            INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                   SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
```

```
//         OUTL=$OUT,                     OUTPUT CLASS OF REPORTS
//         OLDPE=,                              DS NAME OF OLD PE
//         UWK=$UWK,                                   WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)'          TRANSACTION SPACE
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PE25MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGY   DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU920 EXEC PGM=BVPTU920
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MB   DD DSN=&&PE25MB,DISP=(OLD,DELETE)
//PAC7EN   DD DSN=&OLDPE,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7ET   DD SYSOUT=&OUT
//PAF900 EXEC PGM=BVPAF900,COND=(0,NE,PTU920)
//*-----------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
```

```
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA15 EXEC PGM=BVPACA15,
//             COND=((0,NE,PTU920),(0,NE,PAF900))
//*--------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
```

```
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC   DD DUMMY
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//SYSUDUMP DD SYSOUT=&OUT
```

Retrieval of the Development Database (PC25)

PC25 - Introduction

Principle

This procedure (PC25) retrieves the PC file produced by the backup of the old Development Database in a new PC file format.

Execution conditions

None.

Printed output

This procedure prints a report which indicates the number of Manuals changed into Volumes, the code of the new Development Database and the number of records output by the PC file.

Result

The result of this procedure is a sequential image of the new Development Database format. This new PC file must be used as input to the next required step: the re-organization step.

PC25 - Notes on Data Retrieval

Splitting up of the comment description (-G)

The comment description is split up into several descriptions.

- Comments

  They include the comments and the COBOL alias (-GC).

  Caution

  In the 2.0 or 2.5 release, if the type of documentation line was not adapted to the entity type (ex: a generation line in a Data Element), it will become a comment.

- Generation lines

  They include the G, P, V and Z line types (-GG).
- Generation parameters

  They include the O line type (-GO).
- Error messages management

  They include the C, D, F ,S ,T, U line types (-GE).
- Call of entities via Relations

  They include the R line type (-CR).
- Specificity of the Input Aid entity

  The type on the input aid description determines the type value on the definition, i.e. 'C' for comments, 'G' for generation parameters or 'O' for generation options. The input aid calls are accessible through -GC, -GG or -GO.

  WARNING:

  If there are several type values on the same description in the 2.0 or 2.5 release, an error message is displayed, and the error must be corrected manually.

  There again, if the input aid call is wrongly 'Generated' or 'dialogue option', it will become a comment.

  Important: If in the 2.n release, a line with a type which is not a comment is overridden by a comment line, because of the -G splitting, this override is not transferred to the actual release, it must be done manually in the new Database.

Data structures table type

Data Structure with a table type (G, T, M, N) and a Logical View type (V) do not change. All other types (files...) become the Z type. The Report entity is no longer supported by the Data Structure, thus the J type no longer exists.

Transformation of U type manuals

Manuals are replaced with volumes, their codes are completed with '£IBM'.

Extension data: User entities, Input Aids, Report Layouts

There are no more continuation records for these entities. There is only one index for each main record and one index for each continuation record. This Long data is created to concatenate the information included in the previous

records. This data can be 1,000 characters long. It is split up into several records. Now a single index is created and it points to the first of these records.

PC25 - Input / Processing / Results

A * line with the code of the new Development Database.

This line is optional if the Database code indicated in the 2.5 release can be kept. This Database code must have been defined in the Administration Database.

If you do not specify any Database code, an error message is sent and the procedure cannot be run.

This line must be structured in this way:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	4	bbbb	Code of new Database

PC25 - Description of Steps

Input recognition: PTU001

General processes: PTU911

Code	Physical name	Type	Label
PAC7MC	&OLDPC	Input	Sequential image of the network (old release)
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7MB	&&PC25MB	Input	User input
PAC7PB	&&PC	Output	First data record (length=153)
PAC7PE	&&PE	Output	User Entity Occurrence definition (2.5 release), Report layouts, and Comments (except the calls of Input Aids) (length=193)
PAC7PG	&&PG	Output	Description of Input Aids and Comments including calls of Input Aids. (length=193)
PAC7PL	&&PL	Output	Definition and Description of Volumes, Definition and Description of Manuals (length=193)

Code	Physical name	Type	Label
PAC7PZ	&&PZ	Output	User Entities and description of their Occurrences (2.5 release) (length=193)
PAC7PF	&&PF	Output	Other records (length=153)
PAC7PM	&&PM	Output	Report file (length=62)
PAC7ET		Report	Report only if absence of Database code

Manuals and volumes processing: PTU909

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7PB	&&PC	Input	First data record
PAC7PL	&&PL	Input	Definition and Description of Volumes and Manuals
PAC7PI	&&PI	Output	Sorted and re-formatted Volumes Definitions and Descriptions (length=153)
PAC7PM	&&PM	Input/ Output	Report file
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Comments processing: PTU92A

Code	Physical name	Type	Label
PAC7PG	&&PG	Input	Description of Input Aids and of the call of Input Aids in the Comments
PAC7PM	&&PM	Input/ Output	Report
PAC7PE	&&PH	Output	Description of Input Aids and of the call of Input Aids in Comments (length=193)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Sort of Input aids : PTU92B

Code	Physical name	Type	Label
PAC7PE	&&PH	Input	Description of Input Aids and Comments including calls of Input Aids
PAC7PK	&&PK	Output	Description of Input Aids and Comments including calls of Input Aids (length=193)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Sort of Input aids : PTU92C

Code	Physical name	Type	Label
PAC7PK	&&PK	Input	Description of Input Aids and Comments including calls of Input Aids
PAC7KP	&&KP	Output	Description of Input Aids and Comments including calls of Input Aids (length=193)
PAC7PB	&&PC	Input	First data record
PAC7PD	&&PD	Output	First data record (length=153)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Report layout processing: PTU919

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7PE	&&PE	Input	User Entity Occurrences Definition (2.5 rel.), Report layouts and Comments (except calls of input aids)
PAC7PB	&&PD	Input	First data record

Code	Physical name	Type	Label
PAC7PM	&&PM	Input/ Output	Report file
PAC7ZP	&&EP	Output	User entity Occurrences Definition (2.5 rel.), Report layouts (length=193)
PAC7PO	&&PO	Output	Comments (except the call of Input Aids) (length=153)
PAC7KP	&&KP	Input	Comments (including the calls of Input Aids)
PAC7PD	&&PB	Output	First data record (length=153)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Meta entities processing: PTU912

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7PZ	&&PZ	Input	User Entities (2.5 release)
PAC7PB	&&PC	Input	First data record
PAC7ZP	&&ZP	Output	Development Model records (Definition and Descriptions) (length=193)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

User entities processing: PTU913

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7PX	&&EP	Input	Definition of User Entity Occurrences (2.5 release) and Report Layout
PAC7PZ	&&ZP	Input	Definition and Description of the Development Model and Description of User Entity Occurrences (2.5 rel.)

Code	Physical name	Type	Label
PAC7PB	&&PB	Input	First data record
PAC7ZP	&&ZX	Output	Long data of the Development Model, User Entities, Report layouts, and Comments (including the calls of Input Aids) (length=193)
PAC7PD	&&PR	Output	First data record (length=153)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Sort of extension data

Sort criteria: SRTPC25 member of the SY PDS

Code	Physical name	Type	Label
SORTIN	&&ZX	Input	Intermediate extension data
SORTOUT	&&XZ	Output	Sorted long data (length=193)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Files merging: PTU914

This step consists in restoring the final sequential image from the intermediate files produced by the previous steps.

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7ZP	&&XZ	Input	Sorted long data
PAC7PO	&&PO	Input	Comments (no call of Input Aids)
PAC7PD	&&PR	Input	First data record
PAC7PI	&&PI	Input	Volumes Definition and Description
PAC7PF	&&PF	Input	Other records
PAC7PM	&&PM	Input	Report file

Code	Physical name	Type	Label
PAC7PC	&INDUN..&BASE.PC(+1)	Output	Sequential image of the network (present release)
PAC7ET		Report	Retrieval report

PC25 - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*                 RETRIEVAL OF PC FILE
//*
//* ------------------------------------------------------
//*
//BVPPC25  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//         INDSV='$INDSV',              INDEX OF SYSTEM VSAM FILES
//         INDUN='$INDUN',            INDEX OF USER NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         SORTLIB='$BIBT',                          SORT LIBRARY
//         DSCB='$DSCB',                           DSCB MODEL FILE
//         OUT=$OUT,                                 OUTPUT CLASS
//         VOLS='SER=$VOLUN',          VOLUME OF ARCHIVED JOURNAL
//         UNITS=$UNITUN,         BACKUP UNIT (DISK OR CARTRIDGE)
//         OLDPC=,                                DS NAME OF OLD PC
//         UWK=$UWK,                                     WORK UNIT
//         SPAPC='(TRK,(300,10),RLSE)',   DEVPT DATABASE BACKUP 2
//         CYL='(10,1)'                             TEMPORARY SPACE
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PC25MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PTU911 EXEC PGM=BVPTU911
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7MB   DD DSN=&&PC25MB,DISP=(OLD,DELETE)
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MC   DD DSN=&OLDPC,DISP=SHR
//PAC7PF   DD DSN=&&PF,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
```

```
//PAC7PB   DD DSN=&&PC,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(TRK,1,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PE   DD DSN=&&PE,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PG   DD DSN=&&PG,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PL   DD DSN=&&PL,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PM   DD DSN=&&PM,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=62,BLKSIZE=6200)
//PAC7PZ   DD DSN=&&PZ,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7ET   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PTU909 EXEC PGM=BVPTU909,COND=(0,NE,PTU911)
//*----------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7PB   DD DSN=&&PC,DISP=(OLD,PASS)
//PAC7PL   DD DSN=&&PL,DISP=(OLD,DELETE)
//PAC7PI   DD DSN=&&PI,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PM   DD DSN=&&PM,DISP=(OLD,PASS)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PTU92A EXEC PGM=BVPTU92A,COND=(0,NE,PTU911)
//*----------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//PAC7PG   DD DSN=&&PG,DISP=(OLD,DELETE)
//PAC7PM   DD DSN=&&PM,DISP=(OLD,PASS)
//PAC7PE   DD DSN=&&PH,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
```

```
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PTU92B EXEC PGM=BVPTU92B,COND=(0,NE,PTU911)
//*----------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUX   DD SYSOUT=&OUT
//PAC7PE   DD DSN=&&PH,DISP=(OLD,DELETE)
//PAC7PK   DD DSN=&&PK,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PTU92C EXEC PGM=BVPTU92C,COND=(0,NE,PTU911)
//*----------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUX   DD SYSOUT=&OUT
//PAC7PB   DD DSN=&&PC,DISP=(OLD,PASS)
//PAC7PK   DD DSN=&&PK,DISP=(OLD,DELETE)
//PAC7PD   DD DSN=&&PD,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(TRK,1,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7KP   DD DSN=&&KP,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PTU919 EXEC PGM=BVPTU919,COND=(0,NE,PTU911)
//*----------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7KP   DD DSN=&&KP,DISP=(OLD,DELETE)
//PAC7PB   DD DSN=&&PD,DISP=(OLD,DELETE)
//PAC7PE   DD DSN=&&PE,DISP=(OLD,DELETE)
//PAC7PM   DD DSN=&&PM,DISP=(OLD,PASS)
//PAC7PD   DD DSN=&&PB,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
```

```
//PAC7PO   DD DSN=&&PO,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7ZP   DD DSN=&&EP,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PTU912 EXEC PGM=BVPTU912,COND=(0,NE,PTU911)
//*---------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7PZ   DD DSN=&&PZ,DISP=(OLD,DELETE)
//PAC7ZP   DD DSN=&&ZP,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PB   DD DSN=&&PC,DISP=(OLD,DELETE)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//PTU913 EXEC PGM=BVPTU913,COND=(0,NE,PTU911)
//*---------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7PB   DD DSN=&&PB,DISP=(OLD,DELETE)
//PAC7PD   DD DSN=&&PR,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PZ   DD DSN=&&ZP,DISP=(OLD,DELETE)
//PAC7PX   DD DSN=&&EP,DISP=(OLD,DELETE)
//PAC7ZP   DD DSN=&&ZX,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP DD SYSOUT=&OUT
//SORTZX EXEC PGM=SORT,COND=(0,NE,PTU911)
//*-------------------------------------
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
```

```
//SYSPRINT DD SYSOUT=&OUT
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTIN   DD DSN=&&ZX,DISP=(OLD,DELETE)
//SORTOUT  DD DSN=&&XZ,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//SYSIN    DD DSN=&INDSN..BVPSY(SRTPC25),DISP=SHR
//PTU914 EXEC PGM=BVPTU914,COND=(0,NE,PTU911)
//*---------------------------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7PF   DD DSN=&&PF,DISP=(OLD,DELETE)
//PAC7PD   DD DSN=&&PR,DISP=(OLD,DELETE)
//PAC7PI   DD DSN=&&PI,DISP=(OLD,DELETE)
//PAC7PM   DD DSN=&&PM,DISP=(OLD,DELETE)
//PAC7PO   DD DSN=&&PO,DISP=(OLD,DELETE)
//PAC7ZP   DD DSN=&&XZ,DISP=(OLD,DELETE)
//PAC7PC   DD DSN=&INDUN..&BASE.PC(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,
//            VOL=&VOLS,
//            SPACE=&SPAPC,
//            DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//PAC7ET   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

Generation-Print Commands Retrieval (PG20)

PG20 - Introduction

Principle

The PG20 procedure retrieves the 2.0 release PG file, sequential image of the generation-print commands, in the new release format.

It updates the Development Database with the generation-print commands and the Administration Database with the JCL command lines (displayed on the GP screen with the C4 option in the 2.0 release).

Execution conditions

The files of the Administration and Development Databases must be closed in the on-line mode.

Printed output

This procedure outputs a report which contains the errors encountered.

Note

The insertion of update transactions is possible only in libraries or sessions already defined in the Database, otherwise they are rejected.

The PG file may contain commands associated with a specific library or session which can be purged later.

The update of a generation-print command associated with an entity is not possible if the entity is not defined.

Example: for the GCP PROGRA command, the PROGRA program must be defined in the Database.

User codes present in the PG file and not present in the Administration Database are automatically created for users who have JCLs.

PG20 - Input / Processing / Results

A * line with the user code, password and the code of the Development Database for which the JCL command lines were previously updated in the Administration Database.

If you do not specify the user code or the Database code, an error message is sent and the procedure cannot be run.

The line structure is as follows:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	4	cccc	Database code

PG20 - Description of Steps

Input recognition: PTU001

Generation-print commands formatting: PTU908

Code	Physical name	Type	Label
PAC7IN	&OLDPG	Input	Generation-print commands, old release

Code	Physical name	Type	Label
PAC7OU	&&PG	Output	Re-formatted generation-print commands (length=150)

Generation-print commands processing: PTU921

Code	Physical name	Type	Label
PAC7PG	&&PG	Input	Generation-print commands, old release
PAC7AE	&INDSV..BVPAE	Input	Error labels
PAC7MB	&&PG20MB	Input	User Entities
PAC7GY	&&PACGY	Output	Generation-print commands transactions (length=310)
PAC7GZ	&&PACGZ	Output	JCL lines transactions (length = 310)
PAC7ET		Report	Error report

Transactions formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV.. &BASE.AN	Input	Development Database index
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGY	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Formatted transactions (It should be able to contain all input transactions and the elementary deletion transactions which are generated by the multiple deletion transactions) (length=170)
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)

Update of the Development Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Output	Development Database Data file
PAC7AN	&INDUV..&BASE.AN	Output	Development Database index
PAC7AY	&INDUV..&BASE.AY	Output	Development Database extension
PAC7AJ	&INDUV..&BASE.AJ	Output	Development Database journal
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database Elements
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Input	Administration Database data
PAC7AN	&INDSV..BVPGN	Input	Administration Database index
PAC7AE	&INDSV..BVPAE	Input	Error labels

Code	Physical name	Type	Label
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Output	Administration Database Data file
PAC7AN	&INDSV..BVPGN	Output	Administration Database Index file
PAC7AY	&INDSV..BVPGY	Output	Administration Database extension
PAC7AJ	&INDSV..BVPGJ	Output	Administration Database journal
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)

Code	Physical name	Type	Label
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of transactions specific to a user is preceded by a banner with this user's code.

Return codes:

- 0 : OK without error
- 2 : warning error
- 4 : fatal error

PG20 - Execution JCL

```
//* -----------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -----------------------------------------------------
//*             RETRIEVAL OF PG FILE SINCE 2.0
//*
//* -----------------------------------------------------
//*
//BVPPG20  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUTL=$OUT,                     OUTPUT CLASS OF REPORTS
//         OLDPG=,                              DS NAME OF OLD PG
//         UWK=$UWK,                                    WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)'          TRANSACTION SPACE
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PG20MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
```

```
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGY   DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU908 EXEC PGM=BVPTU908
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7IN   DD DSN=&OLDPG,DISP=SHR
//PAC7OU   DD DSN=&&PG,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(10,5),RLSE),
//            DCB=(RECFM=FB,LRECL=150,BLKSIZE=6150)
//PTU921 EXEC PGM=BVPTU921
//*-----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MB   DD DSN=&&PG20MB,DISP=(OLD,DELETE)
//PAC7PG   DD DSN=&&PG,DISP=(OLD,DELETE)
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7GZ   DD DSN=&&PACGZ,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//PAF90Y EXEC PGM=BVPAF900,COND=(0,NE,PTU921)
//*---------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
```

```
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA1Y EXEC PGM=BVPACA15,COND=((0,NE,PTU921),(0,NE,PAF90Y))
//*-------------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC   DD DUMMY
```

```
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//SYSUDUMP DD SYSOUT=&OUT
//PAF90Z EXEC PGM=BVPAF900,COND=(0,NE,PTU921)
//*---------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=10','BUFNI=10')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=10')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&&PACGZ,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA1Z EXEC PGM=BVPACA15,COND=((0,NE,PTU921),
//            (0,NE,PAF90Z))
//*-----------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
```

```
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=10','BUFNI=10')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=10')
//AYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=10')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC   DD DUMMY
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//SYSUDUMP DD SYSOUT=&OUT
```

Generation-Print Commands Retrieval (PG25)

PG25 - Introduction

Principle

The PG25 procedure retrieves the 2.5 release PG file, sequential image of the generation-print commands, in the new format.

It updates the Development Database with the generation-print commands and the Administration Database with the JCL command lines (displayed on the GP screen with the C4 option in the 2.5 release).

Execution conditions

The files of the Administration and Development Databases must be closed to on-line use.

Printed output

This procedure prints a report on the errors encountered.

Note

The insertion of update transactions is possible only in libraries or sessions already defined in the Database, otherwise they are rejected.

The PG file may contain commands associated with a specific library or session which can be purged later.

The update of a generation-print command associated with an entity is not possible if the entity is not defined. Example: for the GCP PROGRA command, the PROGRA program must be defined in the Database.

Any user having JCL lines to generate in online mode (GP screen, displayed in C4 option) is automatically created in the Administration Database.

PG25 - Input / Processing / Results

A * line with the user code, password and the code of the Development Database for which the JCL command lines were previously updated in the Administration Database.

If you do not specify the user code or the Database code, an error message is sent and the procedure cannot be run.

The line structure is as follows:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
22	4	cccc	Database code

PG25 - Description of Steps

Input recognition: PTU001

Generation-print commands processing: PTU921

Code	Physical name	Type	Label
PAC7PG	&OLDPG	Input	Generation-print commands, old release
PAC7AE	&INDSV..BVPAE	Input	Error labels
PAC7MB	&&PG25MB	Input	User Entities
PAC7GY	&&PACGY	Output	Generation-print commands transactions (length=310)
PAC7GZ	&&PACGZ	Output	JCL lines transactions (length = 310)
PAC7ET		Report	Error report

Transactions formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV.. &BASE.AN	Input	Development Database index
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGY	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Formatted transactions (It should be able to contain all input transactions and the elementary deletion transactions which are generated by the multiple deletion transactions) (length=170)
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)

Update of the Development Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Output	Development Database Data file
PAC7AN	&INDUV..&BASE.AN	Output	Development Database index
PAC7AY	&INDUV..&BASE.AY	Output	Development Database extension
PAC7AJ	&INDUV..&BASE.AJ	Output	Development Database journal
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database Elements

Code	Physical name	Type	Label
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Input	Administration Database data
PAC7AN	&INDSV..BVPGN	Input	Administration Database index
PAC7AE	&INDSV..BVPAE	Input	Error labels
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Output	Administration Database Data file
PAC7AN	&INDSV..BVPGN	Output	Administration Database Index file
PAC7AY	&INDSV..BVPGY	Output	Administration Database extension
PAC7AJ	&INDSV..BVPGJ	Output	Administration Database journal
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of transactions specific to a user is preceded by a banner with this user's code.

Return codes:

- 0 : OK without error
- 2 : warning error
- 4 : fatal error

PG25 - Execution JCL

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

```
//* ------------------------------------------------------
//*             RETRIEVAL OF PG FILE SINCE 2.5
//*
//* ------------------------------------------------------
//*
//BVPPG25  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',             DEVPT SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUTL=$OUT,                     OUTPUT CLASS OF REPORTS
//         OLDPG=,                              DS NAME OF OLD PG
//         UWK=$UWK,                                 WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)'          TRANSACTION SPACE
//*********************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PG25MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGY   DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU921 EXEC PGM=BVPTU921
//*-----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
```

```
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MB   DD DSN=&&PG25MB,DISP=(OLD,DELETE)
//PAC7PG   DD DSN=&OLDPG,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7GZ   DD DSN=&&PACGZ,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//PAF90Y EXEC PGM=BVPAF900,COND=(0,NE,PTU921)
//*-----------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA1Y EXEC PGM=BVPACA15,COND=((0,NE,PTU921),(0,NE,PAF90Y))
//*-------------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
```

```
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC   DD DUMMY
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//SYSUDUMP DD SYSOUT=&OUT
//PAF90Z EXEC PGM=BVPAF900,COND=(0,NE,PTU921)
//*---------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=10','BUFNI=10')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=10')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&&PACGZ,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
```

```
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA1Z EXEC PGM=BVPACA15,COND=((0,NE,PTU921),
//             (0,NE,PAF90Z))
//*-----------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=10','BUFNI=10')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=10')
//AYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=10')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC   DD DUMMY
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//SYSUDUMP DD SYSOUT=&OUT
```

Retrieval of PJ Transactions (PJ25)

PJ25 - Introduction

Principle

This procedure (PJ25) is used to change the PJ file, which is a journal file (transactions sequential file), into a new archive file in the new version format.

Execution conditions

None.

Printed output

This procedure generates a transaction file which indicates the number of transactions retrieved in their initial 2.n format the number of transactions converted in the new version format and the number of written transactions.

WARNING: The number of written transactions can be very higher than the read transactions number. From an old 2.n transaction, many transactions of the new version can be created; in particular for the meta entities and user entities.

Result

This procedure generates a PJ journal file in the new release format.

Comments

This conversion process of the journal is optional. It should be executed if required by the batch procedures (Use of Pac/Transfer).

This retrieval procedure must be used only for a conversion from a 2.0 or 2.5 version into the new version.

To retrieve some transactions, it is sometimes necessary to search for information in the new version Database. But the corresponding data may no longer exist in the new Database (example: session or library deleted). In such a case, the old transaction is retrieved with its 2.n format.

PJ25 - Description of Steps

Processing of PJ transactions sequential file: PTU918

Code	Physical name	Type	Label
PAC7PJ	&OLDPJ	Input	journal file old version
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV.. &BASE.AN	Input	Development Database index
PAC7AY	&INDUV..&BASE.AY	Input	Development Database extension data
PAC7JP	&&NEWPJ	Output	Journal file in the new format (length=170)

Code	Physical name	Type	Label
PAC7ET		Report	Report

PJ25 - Execution JCL

```
//* -----------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -----------------------------------------------------
//*                 RETRIEVAL OF PJ FILE
//*
//* -----------------------------------------------------
//*
//BVPPJ25  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',             DEVPT SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OLDPJ=,                              DS NAME OF OLD PJ
//         UWK=$UWK                                    WORK UNIT
//*****************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//PTU918 EXEC PGM=BVPTU918
//*-----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
```

```
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//PAC7PJ   DD DSN=&OLDPJ,DISP=SHR
//PAC7JP   DD DSN=&&NEWPJ,DISP=(,CATLG),UNIT=&UWK,
//            SPACE=(TRK,(300,80),RLSE),
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=17000)
//PAC7ET   DD SYSOUT=&OUT
```

PEI Retrieval (PP25)

PP25 - Introduction

Principle

This procedure retrieves the 2.0 or 2.5 PP file, which is the sequential image of the Production Environment Interface, and updates the Development Database of the installed version.

Execution conditions

The Development Database files must be closed in the on-line mode.

Printed output

This procedure outputs a report which indicates the error encountered.

Result

The procedure generates a transaction file which contains the existing production environments, the list of the generated entities, the default environments (-GO of the Library), the list of production sessions in the new format, and updates the Development Database of the installed version.

Note

Any update transactions in a session or library which is not already defined in the Database will be rejected.

The PP file may contain environments with library codes or sessions to be created or purged later in the 2.0 or 2.5 Database.

When creating environments in the oldest Session (0001 or the Library initialization Session), a default Application is automatically created in the same context.

PP25 - Input / Processing / Results

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

If the user code is not indicated, an error message is displayed and the procedure cannot be run.

The structure of the line is presented as follows:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Option to update the environments
			in inter-library

PP25 - Description of Steps

Input recognition: PTU001

Management of production environment: PTU923

Code	Physical name	Type	Label
PAC7PP	&OLDPP	Input	Back up of production environment (old release)
PAC7AE	&INDSV..BVPAE	Input	Error message
PAC7MB	&&PP25MB	Input	User input
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV.. &BASE.AN	Input	Development Database index
PAC7AY	&INDUV..&BASE.AY	Input	Development Database extension data
PAC7GY	&&PACGY	Output	Records of production environments (length=310)
PAC7ET		Report	Report in case of error

Transactions formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV.. &BASE.AN	Input	Development Database index
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGR	&INDSV..BVPGR	Input	Administration Database data

Code	Physical name	Type	Label
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGY	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Formatted transactions (It should be able to contain all input transactions and the elementary deletion transactions which are generated by the multiple deletion transactions) (length=170)
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)

Update of the Development Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Output	Development Database Data file
PAC7AN	&INDUV..&BASE.AN	Output	Development Database index
PAC7AY	&INDUV..&BASE.AY	Output	Development Database extension
PAC7AJ	&INDUV..&BASE.AJ	Output	Development Database journal
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database Elements
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)

Code	Physical name	Type	Label
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

PP25 - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*             RETRIEVAL OF PP FILE
//*
//* ------------------------------------------------------
//*
//BVPPP25  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUTL=$OUT,                     OUTPUT CLASS OF REPORTS
//         OLDPP=,                              DS NAME OF OLD PP
//         UWK=$UWK,                                    WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)'          TRANSACTION SPACE
//*************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PP25MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
```

```
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGY   DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU923 EXEC PGM=BVPTU923
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//AELSR    DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AE   DD SUBSYS=(&LSR,'DDNAME=AELSR')
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//PAC7MB   DD DSN=&&PP25MB,DISP=(OLD,DELETE)
//PAC7PP   DD DSN=&OLDPP,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//PAF900 EXEC PGM=BVPAF900,COND=(0,NE,PTU923)
//*---------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
```

```
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA15 EXEC PGM=BVPACA15,COND=((0,NE,PTU923),(0,NE,PAF900))
//*-------------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC   DD DUMMY
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//SYSUDUMP DD SYSOUT=&OUT
```

Retrieval of passwords (UTMP)

UTMP - Introduction

Principle

The purpose of this procedure is to retrieve the 2.n user passwords, using as input the PE file produced by the backup of users parameters (PARM procedure), and to integrate them into the Administration Database users file.

Execution conditions

The users file of the Administration Database must be closed to online use.

Printed report

In case of a wrong creation, the procedure prints an error message in display format.

Result

The procedure integrates 2.0 or 2.5 user passwords into the Administration Database.

UTMP - Description of steps

2.N passwords retrieval: PTURMP

Code	Physical name	Type	Label
PAC7CE	&OLDPE	Input	Old version user parameters
PACGGU	&INDSV..BVPGU	Input	Administration Database users

UTMP - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*                 PASSWORD RETRIEVAL
//*
//* ------------------------------------------------------
//*
//BVPUTMP  PROC OUT=$OUT,                          OUTPUT CLASS
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         OLDPE=                               DS NAME OF OLD PE
```

```
//*************************************************************
//PTURMP EXEC PGM=BVPTURMP
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7CE   DD DSN=&OLDPE,DISP=SHR
```

Retrieval of Pac/Transfer Parameters (UV25)

UV25 - Introduction

Principle

The UV25 procedure retrieves the UV PacTransfer parameters file, 2.0 or 2.5 release, in the new format.

It updates the Administration Database.

Execution conditions

The Administration Database files must be closed to on-line use.

Printed output

This procedure prints a report on the errors encountered.

UV25 - Input / Processing / Results

A '*' line with a user code, a password and the code of the Development Database concerned by the transfers.

If you do not specify the user code or the Database code, an error message is sent and the procedure cannot be run.

The line structure must be as follows:

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

Position	Length	Value	Meaning
22	4	cccc	Database code

UV25 - Description of Steps

Input recognition: PTU001

Processing of transfer parameters: PTU922

Code	Physical name	Type	Label
PAC7UV	&OLDUV	Input	Transfer parameters, old release
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7MB	&&UV25MB	Input	User input
PAC7GY	&&PACGY	Output	Transfer parameter transactions (length=310)
PAC7ET		Report	Report in case of error

Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Input	Administration Database data
PAC7AN	&INDSV..BVPGN	Input	Administration Database index
PAC7AE	&INDSV..BVPAE	Input	Error labels
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	&&PAC7ME	Output	Work file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDSV..BVPGR	Output	Administration Database Data file
PAC7AN	&INDSV..BVPGN	Output	Administration Database Index file
PAC7AY	&INDSV..BVPGY	Output	Administration Database extension
PAC7AJ	&INDSV..BVPGJ	Output	Administration Database journal
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database Index file
PACGGR	&INDSV..BVPGR	Input	Administration Database Data file
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	&&PAC7ME	Input	Work file
PAC7MV	&&PAC7MV	Input	Update transactions
PAC7RB	DUMMY	Output	UPDT erroneous transactions (length=80)
PAC7RY	DUMMY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of transactions specific to a user is preceded by a banner with this user's code.

Return codes:

- 0 : OK without error
- 2 : warning error
- 4 : fatal error

UV25 - Execution JCL

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

```
//* ------------------------------------------------------
//*                 RETRIEVAL OF UV FILE
//*
//* ------------------------------------------------------
//*
//BVPUV25  PROC OUT=$OUT,                              OUTPUT CLASS
//         INDSV='$INDSV',              INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OUTL=$OUT,                     OUTPUT CLASS OF REPORTS
//         OLDUV=,                              DS NAME OF OLD UV
//         UWK=$UWK,                                  WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)'          TRANSACTION SPACE
//*************************************************************
//INPUT  EXEC PGM=BVPTU001
//*-----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&UV25MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGY   DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU922 EXEC PGM=BVPTU922
//*-----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MB   DD DSN=&&UV25MB,DISP=(OLD,DELETE)
//PAC7UV   DD DSN=&OLDUV,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
```

```
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//PAF900 EXEC PGM=BVPAF900,COND=(0,NE,PTU922)
//*---------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PACA15 EXEC PGM=BVPACA15,
//             COND=((0,NE,PTU922),(0,NE,PAF900))
//*-------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
```

```
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7DC   DD DUMMY
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//SYSUDUMP DD SYSOUT=&OUT
```

MB Transactions exchanges between 2.n & 3.n (MB25)

MB25 - Introduction

Principle

This procedure retrieves the 2.0 or 2.5 UPDT format transactions in the new release.

Limits

The User Entities, User Entity Occurrences, Relations, U-type Volumes are not processed.

The P.I.A. calls are processed as comments.

The P.I.A. types change between the 2.n and 3.n versions: to allow the exchange of transactions between these 2 versions, execute, in the 2.n version, the UTFG procedure which assigns the types in the 3.n version.

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

This procedure generates a transaction file for the new version UPDT procedure and a revoked transactions file.

MB25 - Description of Steps

MB file retrieval: PTU926

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7MB	&OLDMB	Input	2.5 MB transactions
PAC7MV	&&PACMV	Output	Retrieval transactions for UPDT
PAC7ME	&&PACME	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Batch procedure authorization option

MB25 - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*                 RETRIEVAL OF MB FILE
//*
//* -------------------------------------------------------
//*
//BVPMB25  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',              DEVPT SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OLDMB=,                              DS NAME OF OLD MB
//         UWK=$UWK                                    WORK UNIT
//*************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
```

```
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU926 EXEC PGM=BVPTU926
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7MB   DD DSN=&OLDMB,DISP=SHR
//PAC7MV   DD DSN=&&PACMV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7ME   DD DSN=&&PACME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7EF   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
```

GY Transactions exchanges between 2.n & 3.n (GY25)

GY25 - Introduction

Principle

This procedure retrieves the 2.0 or 2.5 UPDP format transactions in the new release.

Limits

The User Entities, Relations, U-type Manuals are not processed.

The P.I.A. calls are processed as comments.

The P.I.A. types change between the 2.n and 3.n versions: to allow the transactions exchange between these versions, execute in 2.n the UTFG procedure which assigns the 3.n types.

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

This procedure generates a transaction file for the UPDP procedure of the new release and a revoked transactions file.

GY25 - Description of Steps

GY file retrieval: PTU927

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&OLDGY	Input	2.5 GY transactions
PAC7MV	&&PACMV	Output	Retrieval transactions for UPDP
PAC7ME	&&PACME	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

GY25 - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*                 RETRIEVAL OF GY FILE
//*
//* ------------------------------------------------------
//*
//BVPGY25  PROC BASE=$BASE,               CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',              INDEX OF USER VSAM FILES
//         INDSV='$INDSV',            INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
```

```
//*:       SYSTCAT='$SCAT',             DEVPT SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OLDGY=,                              DS NAME OF OLD GY
//         UWK=$UWK                                     WORK UNIT
//*************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU927 EXEC PGM=BVPTU927
//*-----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&OLDGY,DISP=SHR
//PAC7MV   DD DSN=&&PACMV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ME   DD DSN=&&PACME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7EF   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
```

MB Transactions exchanges between 3.n & 2.n (MB30)

MB30 - Introduction

Principle

This procedure retrieves the UPDT format transactions of the new release to the 2.0 or 2.5 release.

Limits

The User Entities, Meta entities, Relations, Descriptions, Keywords, associated Documentation and Relation calls are not processed.

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

This procedure generates a transaction file for the 2.0 or 2.5 UPDT procedure and a revoked transactions file.

MB30 - Description of Steps

MB file retrieval: PTU928

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7MB	&OLDMB	Input	MB transactions of the new version
PAC7MV	&&PACMV	Output	Retrieval transactions for UPDT
PAC7ME	&&PACME	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

MB30 - Execution JCL

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

```
//*                RETRIEVAL OF MB FILE
//*
//* ------------------------------------------------------
//*
//BVPMB30  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',              DEVPT SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OLDMB=,                              DS NAME OF OLD MB
//         UWK=$UWK                                    WORK UNIT
//*************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU928 EXEC PGM=BVPTU928
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7MB   DD DSN=&OLDMB,DISP=SHR
//PAC7MV   DD DSN=&&PACMV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7ME   DD DSN=&&PACME,DISP=(,PASS),UNIT=&UWK,
```

```
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7EF   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
```

GY Transactions exchanges between 3.n & 2.n (GY30)

GY30 - Introduction

Principle

This procedure retrieves the UPDP format transactions of the new release in the 2.0 or 2.5 release.

Limits

The Meta entities, Relations, Descriptions, Keywords, associated Descriptions and Relation calls are not processed.

But, User entities are retrieved as 2.n User entity occurrences. The user will have to create the corresponding 2.n user entity.

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

This procedure generates a transaction file for the 2.0 or 2.5 UPDP procedure and a revoked transactions file.

GY30 - Description of Steps

GY file retrieval: PTU929

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGU	&INDSV..BVPGU	Input	Administration Database users

Code	Physical name	Type	Label
PAC7GY	&OLDGY	Input	GY transactions of the new version
PAC7MV	&&PACMV	Output	Retrieval transactions for UPDP
PAC7ME	&&PACME	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

GY30 - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*                 RETRIEVAL OF GY FILE
//*
//* ------------------------------------------------------
//*
//BVPGY30  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         OUT=$OUT,                                OUTPUT CLASS
//         INDUV='$INDUV',              INDEX OF USER VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//*:       VSAMCAT='$VCAT',                   USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',             DEVPT SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//         OLDGY=,                              DS NAME OF OLD GY
//         UWK=$UWK                                    WORK UNIT
//*****************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//PTU929 EXEC PGM=BVPTU929
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
```

```
//SYSPRINT DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7GY   DD DSN=&OLDGY,DISP=SHR
//PAC7MV   DD DSN=&&PACMV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ME   DD DSN=&&PACME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7EF   DD SYSOUT=&OUT
//PAC7DD   DD SYSOUT=&OUT
```

Procedures - Summary Table of Changes

List of new procedures since the 2.5

Procedure	Comments
ARAD	Archiving Administration Database
ARPM	Archiving the QJ journal of SCM module
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
INQJ	Initialization of SCM Tools Interface archive journal
GPMC	Generation of MOVE CORRESPONDING
GPRC	Generation (COBOL API)
ROAD	Re-organization of Administration Database
RSAD	Restoration of Administration Database
PACG	Backup of Administration Database
PAGX	Extraction of Administration Database
VING	Update of Administration Model
PACS	Management of Development Database
MLIB	Management of libraries
SAVE	Backup of Development Database
GY25	Retrieval of GY file for UPDP to 3.n
MB25	Retrieval of MB file for UPDT to 3.n
GY30	Retrieval of GY file for UPDP to 2.n

Procedure	Comments
MB30	Retrieval of MB file for UPDT to 2.n
PC25	Retrieval of Development Database
PE25	Retrieval of user parameters (PE)
PG20	Retrieval of 2.0 generation-print commands (PG)
PG25	Retrieval of 2.5 generation-print commands (PG)
PJ25	Retrieval of 2.n journal
PP25	Retrieval of PEI environment (PP)
UTMP	Retrieval of passwords
UV25	Retrieval of PacTransfer parameters (UV)
UTM1	Conversion of meta entities (step 1)
UTM2	Conversion of meta entities (step 2)
UTU1	Extraction of 'P' lines with 'UNS'
UTU2	Update of 'P' lines with 'UNS'
CHPM	Environment and Database elements check report
CPPM	Comparison between Database and user configuration
EXPM	Extraction of environments
GPPM	Generation report written into QJ
HIPM	Implementation of elements
SIPM	Generation simulation
UBPM	Generation of batch transactions from QJ
TRED	PacTransfer: print of parameters
UPGP	PAF update of Administration Database
CEND	Intra-Endevor integrity validation
CIND	Inter-Environments integrity validation
GPND	Endevor : recognition of parameters
JJND	Endevor : retrieval of archived journal
JRND	Endevor : retrieval of archived journal transactions
LSND	Endevor : Update of library/session table
MEND	Endevor : Formatting of update transactions
MNDO	Endevor : Formatting of update transactions in 2.5
RIND	Endevor : Creation of Infopac elements (retrieval of existing data)
RPND	Endevor : Creation of VA Pacbase elements (retrieval of existing data)
RPTY	Endevor : Retrieval of the types file

Procedure	Comments
RP25	Endevor : user entities retrieval
RRND	Endevor : Retrieval of existing data
TYND	Endevor : Types loading
UPND	Endevor : Import of VA Pacbase elements into Endevor

List of procedures suppressed since 2.5 release

Procedure	Program	Comments
CPSN	PTU850 PTU855	Integrated in PACX procedure
EMSN	PTU810	
MESN	PTU815	
SASN	PTU130 PTU140	Integrated in PACS procedure
UXSR	UTIXSR	Integrated in PACS procedure
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACTI1	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	
RVKE	PREI40 PREI50	
STOP	PTUR00	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	

Procedure	Program	Comments
VPU2	PVA320	
LVBL	PTULVB	
QREO	PTUN00 PTUN10 PTUN40	
RPPG	PTU908	
RPTD	PTAR20	

Retrieval of VisualAge Pacbase 3.0

Operations to be Performed

The 3.5 version can be installed in the same environment as the 3.0 version. This case is considered as a re-installation.

Before, it is recommended to save the Development and Administration Databases (SAVE, ARCH, SVAD, ARAD) in the 3.0 version.

- If this version is installed in the same environment as the 3.0, refer to the 'Re-installation of Server' chapter in this manual.

  After the Administration Model update (VING execution), connect to the Administrator workbench to enter the new access key.
- If this version is installed in a new environment, after the execution of all the steps of the server part installation, the Databases must be restored following the steps:
  - RSAD : restoration of the Administration Database,
  - VING : update of the Administration Model,
  - Connection via the Administration workbench to enter and activate the new access key to the system,
  - REST : restoration of each Development Database,
  - VINS : update of the Development Model for each Development Database.
- In any case, the reorganization of Databases is required (ARCH, SAVE, REOR, REST).

Chapter 8. Components

Server Environment Components

Introduction

One of the purposes of the product is to manage permanent data in either batch or on-line mode, by using two types of resources:

- Libraries in which the programs which make up the system, and the parameters required to run the system, are stored:
  - One On-Line Program library,
  - One Batch Program library,
  - One library for the common parameters and the Administration Database,
  - One parameter library for each Development Database,
  - One library for the batch procedures' JCLs.
- Permanent files, containing the data handled by the programs defined previously. These files can be classified into two categories:
  - System files, which are not linked to a particular Development Database and remain relatively unchanged,
  - Evolving files which are associated to a Development Database and whose volumes vary according to the updates performed.

Security Systems Interface Extension

Sub-programs are used as interface between VA Pac and the site's security system.

For RACF, the sub-program coded BVPSECRA must be installed in an authorized library by copying the module found in the batch load-module library (SBVPMBR8) under the name BVPSECUR.

Program	Security system
BVPSECRA	RACF
BVPTSS	TOPSECRET Batch
BVPTSSC	TOPSECRET CICS

For more information on the operation of this extension, refer to chapter 'Installation of Server Environment', Subchapter 'Other Installations', 'Security Systems Interface'.

On-Line Documentation

Besides the libraries described in the preceding subchapters, the VA PAC system includes the AE file which contains the error messages and on-line documentation.

Characteristics	Value
Size	approximately 50,000 records per language
Organization	VSAM-KSDS
Recsize	80
CI Size	4,096 (data) 1,024 (index)
Key	12 (position 0)
Use	batch and TP
Dsname	&INDSV..BVPAE

Generation Skeletons

The product also includes the following files:

- A generation skeleton file (SA file) used by the Batch generator function for Cobol API.

Characteristics	Value
Size	Approx. 69 records
Organization	VSAM-KSDS
Recsize	4,605
CI Size	5,120 (data) 1,536 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSA

- A generation skeleton file (SC file) used by the Batch generator function.

Characteristics	Value
Size	Approximately 45 records
Organization	VSAM-KSDS
Recsize	3,204
CI Size	3,584 (data) 1,024 (index)
Key	4 (position 0)
Dsname	&INDSV..BVPSC

- A generation skeleton file (SG file) used by the On-Line Systems Development and Database generator functions.

Characteristics	Value
Size	Approx. 450 records
Organization	VSAM-KSDS
Recsize	4,605
CI Size	5,120 (data) 1,536 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSG

- A generation skeleton file (SN file) used by the eBusiness generator function.

Characteristics	Value
Size	Approx. 350 records
Organization	VSAM-KSDS
Recsize	4,605
CI Size	5,120 (data) 1,536 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSN

- A generation skeleton file (SR file), used by the Reverse generator function.

Characteristics	Value
Size	Approx. 25 records
Organization	VSAM-KSDS
Recsize	4,605
CI Size	5,120 (data) 512 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSR

- A generation skeleton file (SP file) used by the PAF function for the generation of extractors.

Characteristics	Value
Size	Approx. 5 records
Organization	VSAM-KSDS
Recsize	4,605
CI Size	5,120 (data) 512 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSP

- A skeleton file (SF file) used by the PAF function for the generation of extractors.

Characteristics	Value
Size	Approx. 3,000 records
Organization	Sequential
Lrecl	119
Utilization	Batch only
Dsname	&INDSN..BVPSF

- A generation skeleton file (SS file), used by the eBusiness generator function.

Characteristics	Value
Size	Approx. 600 records
Organization	VSAM-KSDS
Recsize	4,605
CI Size	5,120 (data) 1,536 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSS

Parameters

System Parameters Library

The VA Pac system parameters library contains all the paramaters of the system utility programs implemented in the batch procedures used for the installation and operation of the Administration and Development Databases, and in the user standard PAF procedures.

Characteristics	Value
Size	Approx. 20 blocks of 6,080 bytes
Organization	PDS
DCB	Recfm=FB, Lrecl=80, Blksize=6,080
Dsname	&INDSN..BVPSY

The information on names and disks (catalog) is initialized according to the parameters entered upon installation.

Caution

This information should not be modified except in particular cases.

Member	Contents or format	Special indication
BLBVPQU	GENERATION DATA GROUP	
DFBVPAE	DELETE/DEFINE AE	
DFBVPGJ	DELETE/DEFINE GJ	
DFBVPGN	DELETE/DEFINE GN	
DFBVPGR	DELETE/DEFINE GR	
DFBVPGS	DELETE/DEFINE GS	
DFBVPGU	DELETE/DEFINE GU	
DFBVPGY	DELETE/DEFINE GY	
DFBVPLB	DELETE/DEFINE LB	Modifiable (5)
DFBVPPA	DELETE/DEFINE PA	Updatable (1)
DFBVPQJ	DELETE/DEFINE QJ	
DFBVPSA	DELETE/DEFINE SA	
DFBVPSC	DELETE/DEFINE SC	
DFBVPSG	DELETE/DEFINE SG	
DFBVPSN	DELETE/DEFINE SN	
DFBVPSP	DELETE/DEFINE SP	
DFBVPSR	DELETE/DEFINE SR	
DFBVPSS	DELETE/DEFINE SS	
DFBVPTR	DELETE/DEFINE TR	
DFBVPTY	DELETE/DEFINE TY	
DFBVPUP	DELETE/DEFINE UP	

Member	Contents or format	Special indication
DFBVPUQ	DELETE/DEFINE UQ	
DFBVPUU	DELETE/DEFINE UU	
DLBVPUU	DELETE	
DFBVPWS	DELETE/DEFINE WS	Modifiable (6)
DFBVPGK	DELETE/DEFINE GK	
DFDSMSDC	DELETE/DEFINE DC	DSMS: Development Database element (2)
DFSYIANA	DEFINE	Updatable
DLSYIANA	DELETE	
DFSYSEXT	DEFINE	Updatable (3)
DLSYSEXT	DELETE	
DFSYSPAF	DEFINE	Updatable (4)
DLSYSPAF	DELETE	
DFSYTRDU	DEFINE	
DLSYTRDU	DELETE	
DFSYTRPF	DEFINE	
DFWKREOR	DEFINE	
DLWKREOR	DELETE	
DFWYREOR	DEFINE	
DLWYREOR	DELETE	
DFWKROAD	DEFINE	
DLWKROAD	DELETE	
DFWYROAD	DEFINE	
DLWYROAD	DELETE	
DFTABTDF	DELETE/DEFINE TD	TABLES: Description
DLPQCE	DELETE	
KEY01	00000001	
LIBVPGJ	LISTCAT - GJ	
VERIFAE	VERIFY (PAC7AE)	
VERIFAJ	VERIFY (PAC7AJ)	
VERIFAN	VERIFY (PAC7AN)	
VERIFAR	VERIFY (PAC7AR)	
VERIFAY	VERIFY (PAC7AY)	

Member	Contents or format	Special indication
VERIFEM	VERIFY (PAC7EM)	
VERIFGJ	VERIFY (PAC7GJ)	
VERIFGN	VERIFY (PAC7GN)	
VERIFGR	VERIFY (PAC7GR)	
VERIFGU	VERIFY (PAC7GU)	
VERIFGY	VERIFY (PAC7GY)	
VERIFLB	VERIFY (PAC7LB)	
VERIFQJ	VERIFY (PAC7QJ)	
VERIFSA	VERIFY (PAC7SA)	
VERIFSC	VERIFY (PAC7SC)	
VERIFSG	VERIFY (PAC7SG)	
VERIFSN	VERIFY (PAC7SN)	
VERIFSP	VERIFY (PAC7SP)	
VERIFSR	VERIFY (PAC7SR)	
VERIFSS	VERIFY (PAC7SS)	
VERIFTD	VERIFY (PAC7TD)	
SRTPC25	SORT FIELDS	
SRTREO1	SORT FIELDS	
SRTREO2	SORT FIELDS	
MAXKEY	maxi. records	
REPRO	IDCAMS input	
REPRO999	IDCAMS input	

- (1) The size depends on the number of PAF requests sent when using the file in on-line mode.
- (2) must be used if DSMS is to be installed.
- (3) Size depending on the number of requests sent by the deep extractor.
- (4) Size depending on the PAF requests sent in the batch mode when using the file.
- (5) Size depending on the GPRT executions. This file contains the execution reports.
- (6) Size depending on the workstation requests. This work file must contain all the A&D workbench requests for each user.

Batch Procedure JCL Libraries

The procedures must be loaded in a special library called PROCLIB.

This library can be an existing library or one specially created for the purpose.

In some cases, its characteristics must be the following ones:

Characteristics	Value
Size	Approx.150 blocks of 6,080 bytes
Organization	PDS
DCB	Recfm=FB, Lrecl=80, Blksize=6,080
Dsname	user defined.

Administration Database

Administration Database Files

- Data file, GR.

Characteristics	Value
Organization	VSAM-RRDS
Recsize	144
CI size	4,096
Utilization	Batch and on-line
Dsname	&INDSV..BVPGR
Size	27 records per C.I. of 4,096

- Extension data file, GY.

Characteristics	Value
Organization	VSAM-RRDS
Recsize	1,018
CI size	4,096
Utilization	Batch and on-line
Dsname	&INDSV..BVPGY
Size	4 records per C.I. of 4,096

- Index file, GN.

Characteristics	Value
Organization	VSAM-KSDS
Recsize	59
CI size	4,096 (for index and data)
Key	49 (position 0)
Utilization	Batch and TP
Dsname	&INDSV..BVPGN
Size	62 records per C.I. of 4,096

- Journal file, GJ.

Characteristics	Value
Organization	VSAM-RRDS
Recsize	170
CI size	4,096
Utilization	Batch and TP
Dsname	&INDSV..BVPGJ
Size	23 records per C.I. of 4,096

- User file, GU.

Characteristics	Value
Organization	VSAM-RRDS
Recsize	80
CI size	4,096
Utilization	Batch and on-line
Dsname	&INDSV..BVPGU
Size	50 records per C.I. of 4,096

This file is sensitive and its backup must be executed in a secure environment of the installation site.

- Journal file of SCM Tools Interface QJ

Characteristics	Value
Organization	VSAM-RRDS
Recsize	1,105

Characteristics	Value
CI size	1,105
Utilization	Batch and on-line
Dsname	&INDSV..BVPQJ
Size	4 records per C.I. of 4,096

Administration Database Backup

The Administration Database backup consists of two sequential generation files.

- Backup of the Database (PE).

This is a backup file of the Administration Database components: index (GN), Data (GR) and extension (GY) in a sequential format.

Characteristics	Value
Organization	Sequential, generation
Dsname	&INDSN..BVPPE(n)
DCB	Recfm=VB, Lrecl=1,023, Blksize=27,998
Utilization	batch
Size	144 bytes per data
	1,023 bytes per extension data
	59 bytes per index

- Backup of the journal (PK).

The purpose of this file is to store all the update transactions that have affected the Administration Database since its installation and that have passed through the transactions file (GJ).

When the size of this file becomes incompatible with operation requirements, the ARAD procedure enables you to split it into several files, among which only the most recent one is used on a regular basis.

Characteristics	Value
Organization	Sequential, generation
Dsname	&INDSN..BVPPK(n)
DCB	Recfm=FB, Lrecl=170, Blksize=6,800
Utilization	batch

- Backup of SCM module QJ journal

The backup of QJ is the JQ file.

The purpose of this file is to archive the valid transactions already processed by different updates and stored in the QJ journal file.

Characteristics	Value
Organization	Sequential, generation
DSNAME	&INDSN..BVPJQ(n)
DCB	Recfm=FB, Lrecl=1,119, Blksize=1,119
Utilization	batch

Development Database

Development Database Files

These files contain all the data related to development of applications.

- The Data file (AR).

Characteristics	Value
Organization	VSAM-RRDS
Recsize	144
CI size	4,096
Utilization	Batch and TP
Dsname	&INDUV..&BASE.AR
Size	27 records per C.I. of 4,096

- The extension data file (AY).

Characteristics	Value
Organization	VSAM-RRDS
Recsize	1,018
CI size	4,096
Utilization	Batch and On-line
Dsname	&INDUV..&BASE.AY
Size	4 records per C.I. of 4,096

- The index file (AN).

Characteristics	Value
Organization	VSAM-KSDS
Recsize	59
CI size	4,096 (for index and data)
Key	49 (position 0)
Utilization	Batch and On-line
Dsname	&INDUV..&BASE.AN
Size	62 records per C.I. of 4,096

- The journal file (AJ).

All the transactions performed on the Database in batch or on-line mode are saved for two reasons:

- To allow Database restoration if the system standard securities were to fail.
- This information may be used for statistical purposes.

These transactions are usually stored in the transactions backup file (PJ). The transactions file is used temporarily, between the moment transactions are processed by the system and the moment they are saved on their final storage medium by the ARCH procedure.

Characteristics	Value
Organization	VSAM-RRDS
Recsize	170
CI size	4,096
Utilization	Batch and on-line
Dsname	&INDUV..&BASE.AJ
Size	23 records per C.I. of 4,096

Development Database Parameters Library

Characteristics	Value
Size	Approx. 10 blocks of 6,080
Organization	PDS
DCB	Recfm=FB, Lrecl=80, Blksize=6,080
Dsname	&INDUN..&BASE.SY

This library contains the SYSINs of the IDCAMS utility implemented in the batch procedures used for the operations of a Development Database.

The information related to file names and disks (catalog), is initialized according to the initial installation parameters.

- DELETE/DEFINEs of the Database files:

Their names have the DFxxnnff format (xx=ROOT, nn=FILE, and ff=suffix of the relevant file.)

Important note: In these members, the VA Pac Database Administrator can modify the size of the files that make up the Database, according to the evolution of this Database.

- LISTCAT of VSAM file AJ (Development Database Journal):

Its name has the LIxxnnAJ format.

Development Database Backup Files

According to the 'Dispatch' option taken into account upon restoration, the Database backup is either made of two sequential generation files (PC and PJ) or of four sequential generation files (PC, PD, PY and PJ).

- Database or Data backup (PC).

This is a sequential backup file of the Development Database components (Data (AR), Index (AN), and extension (AY) if 'Dispatch' option) or of Data only (AR).

Characteristics	**Value**
Organization	Sequential, generation
Dsname	&INDUN..&BASE.PC(n)
DCB	Recfm=VB, Lrecl=1,023, Blksize=27,998
Utilization	Batch
Size	144 bytes per data
	1,018 per extension data
	59 bytes per index

- The backup of the Development Database index (PD) if 'Dispatch' option.

Characteristics	**Value**
Organization	Sequential, generation
Dsname	&INDUN..&BASE.PD(n)

Characteristics	Value
DCB	Recfm=VB, Lrecl=1,023, Blksize=27,998
Utilization	Batch
Size	59 bytes per index

- The backup of the Development Database random data (PY) if 'Dispatch' option.

Characteristics	Value
Organization	Sequential, generation
Dsname	&INDUN..&BASE.PY(n)
DCB	Recfm=VB, Lrecl=1,023, Blksize=27,998
Utilization	Batch
Size	1,018 bytes per index

- Journal backup (PJ)

The purpose of this file is to store all the update transactions performed in the Development Database since its installation, and that have passed through the transactions file (AJ).

When the size of this file becomes incompatible with operation requirements, the ARCH procedure enables you to split it into several files, among which only the most recent one is used on a regular basis.

Characteristics	Value
Organization	Sequential, generation
Dsname	&INDUN..&BASE.PJ(n)
DCB	Recfm=FB, Lrecl=170, Blksize=27,880
Utilization	batch

Modules - Specific Files

Pac/Impact:

- File of already-impacted criteria (FQ).

Characteristics	Value
Organization	Sequential with generation
DCB	Recfm=FB, Lrecl=100, Blksize=21,600
Dsname	&INDUN..&USER..&BASE.FQ(n)

Characteristics	Value
Utilization	Memorize the impact search criteria that have already been processed

- Search criteria or entry points file (FH).

Characteristics	Value
Organization	Sequential with generation
DCB	Recfm=FB, Lrecl=160, Blksize=24,000
Dsname	&INDUN..&USER..&BASE.FH(n)
Utilization	Memorize impact search criteria for the next IANA execution

- Reduced file of criteria for purge (FR).

Characteristics	Value
Organization	Sequential with generation
DCB	Recfm=FB, Lrecl=72, Blksize=21,600
Dsname	&INDUN..&USER..&BASE.FR(n)
Utilization	Purge the impact search criteria in a text editor.

- Impact result file (FO).

Characteristics	Value
Organization	Sequential with generation
DCB	Recfm=FB, Lrecl=260, Blksize=26,000
Dsname	&INDUN..&USER..&BASE.FO(n)
Utilization	Memorize all the results of the impact analysis.

- File of entities to be analyzed (FP).

Characteristics	Value
Organization	VSAM-KSDS
Recsize	33
CI size	4,096
key	33 (position 0)
Dsname	&INDUV..&USER..&BASE.FP
Utilization	Restrict the impact analysis to the entities specified in the file

DSMS:

When the DSMS function (refer to the 'DSMS' manual) is available on site, a DSMS file is accessed for each Development Database, in batch and on-line modes.

This file contains the list of the entities concerned by each change. The change number is entered by the user on the Database sign-on screen.

- The DSMS file of the Development Database elements (DC)

Characteristics	Value
Organization	VSAM-KSDS
Recsize	mini 50, maxi 168
CI size	4,096
Key	31 (position 2)
Utilization	Accessed by batch and on-line updates
Ddname batch	PAC7DC (Proc. UPDT, REST, RESY)
Ddname CICS	ddy0DC (dd=ROOTD, y=ROOT2 of DSMS installation)

This file is allocated and initialized at the time of the installation of the DSMS Function.

The definition supplied when installing VA Pac must be used if the DSMS has not been installed on the site yet.

PAF:

- PAF work file (PA) for PAF-TP and PUF-TP.

All the user on-line programs which access Databases with the same root need an indexed work file to use the PAF and PUF functions.

Characteristics	Value
Organization	VSAM-KSDS
Recsize	average: 200, maxi: 539
CI size	4,096
Key	37 (position 2)
Utilization	Updated by the BVPTPST and/or BVPTPWS sub-programs, which are called by On-line user programs
Dsname	&INDSV..BVPPA
Ddname	BVPPA

- Work file for PAF in batch mode

All the user batch programs need an indexed work file to use the PAF function. This file is allocated for the job duration and is destroyed at the end of the job.

Characteristics	Value
Organization	VSAM-KSDS
Recsize	average: 170, maxi: 1,031
CI size	4,096
Key	12 (position 0)
Utilization	Updated by the BVPBBT98 sub- program which is called by BVPBBTST/BVPBBTWS (PAF sub- programs called by the user programs).
Dsname	&INDUV..SYSPAF.&USER

PAF extension

Extraction master path file (GS), containing the user's extractors and macro-commands.

Characteristics	Value
Organization	VSAM-KSDS
Recsize	230
CI size	4,096
Key	49 (position 0)
Utilization	Batch and On-line
Dsname	&INDUV..BVPGS

Complementary Libraries and Files

Complementary Dictionary files are automatically downloaded via SMP/E in the hlq.SBVPDIC PDS.

PQC function

The BVPQCRA and BVPQCRF members contain the standard quality rules.

- Specific members of Pacbench Quality Control function

Member	Contents of format	Comments
BVPQCRA	Sequential file in English	Standard rules
BVPQCRF	Sequential file in French	Standard rules

Chapter 9. Appendix

Installation of the Administration Database Model

VING - Introduction

The VING procedure performs the batch update of the Administration Database using transactions provided by IBM. Another file may sometimes be used. For example when the Pacdesign/Pacbench methodology changes.

Execution conditions

The Database must be closed to on-line processing.

Abnormal execution

Refer to the Administrator's Procedures manual, sub-chapter 'Abnormal Ending'.

When an abend occurs during the execution of the BVPACI30 or BVPACI40 program, the Database is no longer consistent.

Once the problem has been solved, the Database must be reloaded with a retrieval of archived transactions and the VING procedure must be executed again.

VING - Input / Processing / Results

This procedure requires two types of user input:

- A line which contains the User ID as well as the operation to perform,
- The transactions which enable the creation of IBM Meta Entities and the retrieval of client User Entities with the 'extension' format: the user should never modify the content of these transactions.

The structure of the line is the following:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Library code
29	4	'VINS'	

Position	Length	Value	Meaning
33	1	'I'	Installation of IBM Meta Entities

Printed output

The procedure outputs:

- a report listing the executed programs,
- the list of requests with the errors detected if any,
- a report of the updates performed by the installation.

Result

Once the update is performed, the network is ready for either on line or batch use.

Note

The extracted transactions to be used by the ROAD procedure are to be written in a file. To do so, take the following line into account (this line is a comment otherwise):

```
'//*VINS.PAC7MR  DD DSN=ROADFILE'.
```

VING - Description of Steps

Input recognition: PTU001

Initialization of the KSDS work file: IDCAMS

Check of VSAM files: IDCAMS

Update of the Administration Database: VING

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error labels
PACGGN	&INDSV..BVPGN	Input	Administration Database Index
PACGGR	&INDSV..BVPGR	Input	Administration Database Data
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database Users
PAC7AJ	&INDSV..BVPGJ	Output	Administration Database Journal
PAC7AN	&INDSV..BVPGN	Output	Administration Database Index
PAC7AR	&INDSV..BVPGR	Output	Administration Database Data

Code	Physical name	Type	Label
PAC7AY	&INDSV..BVPGY	Output	Administration Database Extension
PAC7MA	&FDIC	Input	IBM Meta Entities Transactions
PAC7MB	&&VINGMB	Input	User Input
PAC7BM	&&PACXBM	Input/ Output	Work file
PAC7WD	&&PACXWD	Input/ Output	Extracted Transactions
PAC7ES	&&PACXES	Input/ Output	Extracted Transactions
PAC7TD	&&TD	Input/ Output	Extracted Transactions
PAC7MR	&&MR	Output	Extracted Transactions for ROAD
PAC7MX	DUMMY	Output	Non extracted entities
PAC7IA		Report	General Report of Programs sequence
PAC7EE		Report	Report
PAC7EQ		Report	Report
PAC7EU	DUMMY	Report	Report
PAC7ER		Report	Report
PAC7EZ		Report	Report
PAC7DD		Report	Report
PAC7IE		Report	Report
PAC7IF		Report	Report
PAC7IG		Report	Report
PAC7IH		Report	Report
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0 : No error detected on files
- 4 : Correct the errors and restart the procedure
- 8 : No access authorization for batch procedure
- 12: Input-output error on a file

VING - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*        - DATABASE ADMINISTRATOR
//*        - DICTIONARY UPDATING WITH IBM MODEL ADMIN   -
//* -------------------------------------------------------
//*
//* THE VING PROCEDURE PERFORMS A BATCH UPDATE OF THE DATA
//* BASE ADMIN. , BASED ON TRANSACTIONS PROVIDED.
//*
//* INPUT :
//*  -  USER IDENTIFICATION LINE (REQUIRED)
//*     COL  2 : "*"
//*     COL  3 : USERIDXX
//*     COL 11 : PASSWORD
//*     COL 29 : "VINS"
//*     COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES
//* -------------------------------------------------------
//BVPVING  PROC INDSV='$INDSV',        INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//         INDUV='$INDUV',              INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                 SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         SORTLIB='$BIBT',                          SORT LIBRARY
//         OUT=$OUT,                                 OUTPUT CLASS
//         OUTL=$OUT,                    OUTPUT CLASS OF REPORTS
//         FDIC=,                 DSN USER'S ADMINISTRATION MODEL
//         USER=,                                       USER CODE
//         LSR='BLSR',                     LSR BATCH SYSTEM NAME
//         UWK=$UWK,                                    WORK UNIT
//         SPAMB='(TRK,(5,1),RLSE)',             WORK FILE SPACE
//         SPAWK='(TRK,(50,10))'                 WORK FILE SPACE
//*************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&VINGMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
```

```
//DEFINE EXEC PGM=IDCAMS
//*---------------------
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//MAXKEY  EXEC PGM=IDCAMS
//*----------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//MAXKEY   DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//VERIFY EXEC PGM=IDCAMS
//*---------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGY   DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*
//VINS   EXEC PGM=BVPVINS,REGION=0K
//*------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//ANLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=40')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=40','BUFNI=30')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=40')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=40')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=40')
//SYSEXT   DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//PAC7IA   DD SYSOUT=&OUTL
```

```
//PAC7DD   DD SYSOUT=&OUTL
//PAC7EE   DD SYSOUT=&OUTL
//PAC7EQ   DD SYSOUT=&OUTL
//PAC7ER   DD SYSOUT=&OUTL
//PAC7EU   DD DUMMY
//PAC7EZ   DD SYSOUT=&OUTL
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7IG   DD SYSOUT=&OUTL
//PAC7IH   DD SYSOUT=&OUTL
//PAC7MA   DD DSN=&FDIC,DISP=SHR
//PAC7MB   DD DSN=&&VINGMB,DISP=(OLD,DELETE,DELETE)
//PAC7BM   DD DSN=&&PACXBM,DISP=(,DELETE),UNIT=&UWK,
//            DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7ES   DD DSN=&&PACXES,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7WD   DD DSN=&&PACXWD,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7MR   DD DSN=&&MR,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PAC7MX   DD DUMMY
//PAC7TD   DD DSN=&&TD,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SYSUDUMP DD SYSOUT=&OUT
//*
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-----------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSEXT),DISP=SHR
//PACROU DD DSN=&&DLSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*------------------------------
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DLSYSEXT,DISP=(OLD,DELETE)
```

Installation of the Development Database Model

VINS - Introduction

The VINS procedure performs the batch update of the Development Database using transactions provided by IBM. Another file may sometimes be used (for example when the Pacdesign/Pacbench methodology changes).

Entities are created in inter-Library mode and in the 0001Z session. They can thus be accessed from any Library of the Development Database and from any session.

Execution conditions

The Database must be closed to on-line processing.

Abnormal execution

Refer to the 'Administrator's Procedures' manual, sub-chapter 'Abnormal Ending'.

When an abend occurs during the execution of the BVPACI30 or BVPACI40 programs, the Database is no longer consistent.

Once the problem has been solved, the Database must be reloaded with a retrieval of archived transactions and the VINS procedure must be executed again.

VINS - Input / Processing / Results

This procedure requires two types of user input:

- a line which contains the User ID as well as the operation to perform,
- the transactions which enable the creation of IBM Meta Entities and the retrieval of client User Entities with the 'extension' format: the user should never modify the content of these transactions.

The structure of the line is the following:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Library code
29	4	'VINS'	
33	1	'I'	Installation of IBM Meta Entities

Position	Length	Value	Meaning
		'R'	Retrieval of User Entities with the 'extension' format
		' '	'I' + 'R'

Printed output

The procedure prints

- a report listing the executed programs,
- the list of requests with the errors detected if any,
- a report of the updates performed by the installation,
- a report of the updates performed by the retrieval,

Result

Once the update is performed, the Development Database is ready for either on-line or batch use.

For the retrieval of User Entities with an extension format, a sequential file of purge transactions can be generated (if 'R' in column 33 in the user input). You then have to reorganize the Database with this file as input. The reorganization of the Database with the PC file, saved after the retrieval, is also required if the client user entities of the meta entities exist in more than one session.

Note

Extracted transactions to be used as input to the REOR procedure must be copied in a file by taking the following - otherwise comment - line into account:

```
'//*VINS.PAC7MR  DD DSN=REORFILE'.
```

VINS - Description of Steps

Input recognition: PTU001

Initialization of the KSDS work file: IDCAMS

Check of VSAM files: IDCAMS

Update of the Development Database : VINS

Code	Physical name	Type	Label
PAC7AE	&INDSV..BVPAE	Input	Error Labels

Code	Physical name	Type	Label
PACGGN	&INDSV..BVPGN	Input	Administration Database Index
PACGGR	&INDSV..BVPGR	Input	Administration Database Data
PACGGY	&INDSV..BVPGY	Input	Administration Database Extension
PACGGU	&INDSV..BVPGU	Input	Administration Database Users
PAC7AJ	&INDUV..&BASE.AJ	Output	Development Database Journal
PAC7AN	&INDUV.. &BASE.AN	Output	Development Database Index
PAC7AR	&INDUV..&BASE.AR	Output	Development Database Data
PAC7AY	&INDUV..&BASE.AY	Output	Development Database Extension
PAC7MA	&FDIC	Input	IBM Meta Entities Transactions
PAC7MB	&&VINSMB	Input	User Input
PAC7BM	&&PACXBM	Input/ Output	Work File
PAC7WD	&&PACXWD	Input/ Output	Extracted Transactions
PAC7ES	&&PACXES	Input/ Output	Extracted Transactions
PAC7TD	&&TD	Input/ Output	Extracted Transactions
PAC7MR	&&MR	Output	Extracted Transactions for REOR
PAC7MX	DUMMY	Output	Non extracted entities
PAC7IA		Report	Complete printing of programs sequence
PAC7EE		Report	Report
PAC7EQ		Report	Report
PAC7EU	DUMMY	Report	Report
PAC7ER		Report	Report
PAC7EZ		Report	Report
PAC7DD		Report	Report
PAC7IE		Report	Report
PAC7IF		Report	Report
PAC7IG		Report	Report
PAC7IH		Report	Report
SORTWK01		Sort	

Code	Physical name	Type	Label
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0 : No error detected on files
- 4 : Correct the errors and restart the procedure
- 8 : No access authorization for batch procedure
- 12: Input-output error on a file

Deletion of the KSDS work file: IDCAMS

VINS - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*        - DICTIONARY UPDATING WITH IBM MODEL DEVPT   -
//*
//* -------------------------------------------------------
//*
//* THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
//* DATABASE, BASED ON TRANSACTIONS PROVIDED.
//*
//* INPUT :
//*  -  USER IDENTIFICATION LINE (REQUIRED)
//*     COL  2 : "*"
//*     COL  3 : USERIDXX
//*     COL 11 : PASSWORD
//*     COL 29 : "VINS"
//*     COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES
//*              "R" - RETRIEVAL OF USER ENTITIES WITH THE
//*                    "EXTENSION" FORMAT
//*              " "   "I" + "R"
//* -------------------------------------------------------
//*
//BVPVINS  PROC BASE=$BASE,               CODE OF DEVPT DATABASE
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//         INDUV='$INDUV',               INDEX OF USER VSAM FILES
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                 SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         SORTLIB='$BIBT',                           SORT LIBRARY
//         OUT=$OUT,                                  OUTPUT CLASS
//         OUTL=$OUT,                      OUTPUT CLASS OF REPORTS
//         FDIC=,                    DSN USER'S DEVELOPMENT MODEL
//         USER=,                                       USER CODE
//         LSR='BLSR',                     LSR BATCH SYSTEM NAME
```

```
//         UWK=$UWK,                                        WORK UNIT
//         SPAMB='(TRK,(5,1),RLSE)',                 WORK FILE SPACE
//         SPAWK='(TRK,(50,10))'                     WORK FILE SPACE
//*************************************************************
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&VINSMB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV'
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DEFINE EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//MAXKEY  EXEC PGM=IDCAMS
//*---------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//MAXKEY   DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGY   DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//*
```

```
//VINS   EXEC PGM=BVPVINS,REGION=0K
//*------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AJ   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR')
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=40','BUFNI=30')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=40')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=40')
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=40')
//SYLSR    DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//SYSEXT   DD SUBSYS=(&LSR,'DDNAME=SYLSR')
//PAC7IA   DD SYSOUT=&OUTL
//PAC7DD   DD SYSOUT=&OUTL
//PAC7EE   DD SYSOUT=&OUTL
//PAC7EQ   DD SYSOUT=&OUTL
//PAC7ER   DD SYSOUT=&OUTL
//PAC7EU   DD DUMMY
//PAC7EZ   DD SYSOUT=&OUTL
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7IG   DD SYSOUT=&OUTL
//PAC7IH   DD SYSOUT=&OUTL
//PAC7MA   DD DSN=&FDIC,DISP=SHR
//PAC7MB   DD DSN=&&VINSMB,DISP=(OLD,DELETE,DELETE)
//PAC7BM   DD DSN=&&PACXBM,DISP=(,DELETE),UNIT=&UWK,
//            DCB=BLKSIZE=3440,SPACE=&SPAMB
//PAC7ES   DD DSN=&&PACXES,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//PAC7WD   DD DSN=&&PACXWD,DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK02 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//SORTWK03 DD UNIT=&UWK,SPACE=(CYL,(3,1),,CONTIG)
//PAC7MR   DD DSN=&&MR,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
```

```
//PAC7MX   DD DUMMY
//PAC7TD   DD DSN=&&TD,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAWK,
//            DCB=(RECFM=FB,LRECL=286,BLKSIZE=6292)
//SYSUDUMP DD SYSOUT=&OUT
//*
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER,&INDUV',COND=EVEN
//*-------------------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DLSYSEXT),DISP=SHR
//PACROU DD DSN=&&DLSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//DELETE EXEC PGM=IDCAMS,COND=EVEN
//*------------------------------
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DLSYSEXT,DISP=(OLD,DELETE)
```

Retrieval utilities

UTU1 UTU2 - Adaptation of 'UNS' operators

UTU1 - Extraction of 'P' lines with 'UNS'

Principle

This tool is used to extract the programs' 'P' lines which contain the 'UNS' operator and do not contain anything in the 'Level-Condition type' field. The user will have to verify the output file and delete, in this file, all the lines he/she wants to keep. For the other lines, the 'Level-Condition type' area will have to be set to '99BL' before the UTU2 procedure execution.

Execution conditions

Specify, in the startup JCL, the name of the result file for the lines to be checked: NOMUT='... '

Result

A 'NOMUT' file which contains UNS 'P' lines to check.

UTU1 - Input / Processing / Results

No user input.

UTU1 - Description of Steps

Extracting 'P' lines with 'UNS' operator: UTIUN1

Code	Physical name	Type	Label
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the Development Database
UTUTO	&NOMUT	Output	'P' lines with 'UNS' to be updated

UTU1 - Execution JCL

```
//* -------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* -------------------------------------------------------
//*         EXTRACTION OF LINES "P"
//*           WITH OPERATOR "UNS"
//*           WITHOUT LEVEL-CONDITION TYPE
//* -------------------------------------------------------
//*
//BVPUTU1  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDUN='$INDUN',           INDEX OF USER NOT VSAM FILES
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         OUT=$OUT,                              OUTPUT CLASS
//         VOLS='SER=$VOLUN',                VOLUME FILE &&NOMUT
//         UWK=$UWK,                        WORK UNIT FILE &&NOMUT
//         SPAUT='(TRK,(10,05),RLSE)'          SPACE FILE &&NOMUT
//*************************************************************
//UTIUN1 EXEC PGM=BVPTUNS1
//*-----------------------------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7PC   DD DSN=&INDUN..&BASE.PC(0),DISP=SHR
//UTUTO    DD DSN=&NOMUT,DISP=(,CATLG,DELETE),
//            UNIT=&UWK,VOL=&VOLS,SPACE=&SPAUT,
//            DCB=(RECFM=FB,LRECL=50,BLKSIZE=5000)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

UTU2 - Update of 'P' lines with 'UNS'

Principle

This tool is used to update the transactions extracted with the UTU1 procedure.

For all the lines in the input file, the 'Level- Condition type' area is forced to '99BL'.

Execution conditions

Specify the result file name of the UTU1 procedure in the startup JCL: NOMUT='... '

Result

A new sequential image of the Development Database.

UTU2 - Input / Processing / Results

No user input.

UTU2 - Description of Steps

Lines 'P' update with 'UNS' operator : UTIUN2

Code	Physical name	Type	Label
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the Development Database
UTUTI	&NOMUT	Input	'P' lines with 'UNS' to be updated
PAC7CP	&INDUN..&BASE.PC(+1)	Output	New sequential image of the Development Database

UTU2 - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*           UPDATE LINES "P"
//*           WITH OPERATOR "UNS"
//* ------------------------------------------------------
//*
//BVPUTU2  PROC BASE=$BASE,               CODE OF DEVPT DATABASE
//         INDUN='$INDUN',           INDEX OF USER NOT VSAM FILES
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         OUT=$OUT,                             OUTPUT CLASS
//         DSCB='$DSCB',                      DSCB MODEL FILE
//         VOLS='SER=$VOLUN',                  VOLUME FILE PC
//         UNITS=$UNITUN,                          SAVE UNIT
//         SPAPC='(TRK,(300,10),RLSE)'         SPACE PACBASE BASE
//*************************************************************
//UTIUN2 EXEC PGM=BVPTUNS2
//*-----------------------------------------------------------
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//PAC7PC   DD DSN=&INDUN..&BASE.PC(0),DISP=SHR
```

```
//UTUTI    DD DSN=&NOMUT,DISP=SHR
//PAC7CP   DD DSN=&INDUN..&BASE.PC(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=&UNITS,VOL=&VOLS,SPACE=&SPAPC,
//            DCB=(&DSCB,RECFM=VB,LRECL=1023,BLKSIZE=27998)
//SYSPRINT DD SYSOUT=&OUT
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
```

UTM1 UTM2 - Conversion of 'old' Meta Entities

Principle

This procedure is a utility whose purpose is to convert the 'old' meta entities into 'formatted' meta entities.

The PC25 procedure retrieves the 2.n user entities into 3.n 'old'-type meta entities. These meta entities cannot be modified.

Once converted, the 'formatted' meta entities can be modified and enriched.

You will have to operate manually during this conversion, which consists of several steps.

Migration process

Step 1: Analysis of the Database relations (UTM1)

In the 3.n version, a relation is associated with one entity type and not with several types as in the 2.n version.

Any 'old'-type relation must be converted into as many relations as the number of entities they are associated with.

This first step consists in searching, in the VA Pacbase Database, the calls to 'old' relations in the description lines of the meta entities and in all the entities.

The list of calls is printed in the output file.

In this file, the '&&' characters separate the printed information from the information in transaction format.

You will have to specify the new relation code.

The old code can be kept if the relation is associated with one entity type only.

Example of transformation:

```
RELAT1  P    220  P   PGMXCR  CR   VAO  2243   &&     ...
```

modified as:

```
RELATE  P    220  P   PGMXCR  CR   VAO  2243   &&     ...
```

The RELATE relation will then be dedicated to the 'P'-type entity (Program), whereas RELAT1 was dedicated to several entity types.

Note: Only the Relation name, located before the && characters, can be modified. The continuation of the line must remain as is.

Step 2: Conversion of meta entities, relations and impacted entities and update (UTM2)

- The 'old'-type meta entities are converted into 'formatted' meta entities.
- The relations you have selected are modified or created.

  If the relation already exists in the Database, it can be kept if it is defined in a convenient context for the calling entity, if it is 'constrained' and if the associated entity type is the same as the chosen type.

  If the relation is 'old', it will be changed into 'constrained' and the chosen entity type will be associated with it.

  If the existing relation is not suitable, an error message is printed and you will have to choose a new relation code.

  Any relation creation is performed in session 1 and in inter-library mode.
- Relations calls are updated in the concerned entities (description lines of meta entities and others).
- If errors are detected (return code 8), you will have to modify the input file PAC7ME and start again the UTM2 procedure before the update. Caution, the &UTM2MV permanent file must first be deleted or renamed.

  Update

  The update is performed by the BVPACA15 program.

  A backup of the Database is recommended before the update.

  Only transactions where an error is detected are printed. There may be many rejects because these entities created in the 2.n version are much more controlled in the 3.n version. Manual interventions in the Database are then required.

  The transactions are not journalized.

  The &UTM2MV transaction file is declared as a permanent file to enable the user to view all the transactions impacted by the update.

Step 3: Inventory of fixtures

It is recommended to re-execute the first step to make sure that the Database no longer contains calls to 'old'-type relations.

Otherwise, you will have to operate again and re-execute the next steps.

Step 4: Reorganization

When the conversion is acceptable, you must reorganize the Database.

Execution conditions

None during step 1 (UTM1).

For step 2 (update), the AR, AN, AJ and AY files must be closed to online use (except for platforms which allow a Batch/Online concurrency).

Printed output

At the end of step 1, a report is printed. It contains the list of the calls to 'old'-type relations.

At the end of step 2 before the update, error messages are printed in display format.

At the end of the update, a report lists the errors encountered.

Result

Once the reorganization is performed, the result is a Database free from 'old'-type meta entities and calls to 'old'-type relations.

UTM1 - Description of Steps

Analysis of the Database relations: PTUME1

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV..&BASE.AN	Input	Development Database Index
PAC7ME	&UTM1FILE	Output	List of relations calls in all the entities (length = 221)
PAC7AE	&INDSV..BVPAE	Input	Error Labels
PAC7EQ		Report	Report

UTM1 - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*            - MIGRATION META-ENTITES OLD 1 -
//*
//* ------------------------------------------------------
//BVPUTM1  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDUV='$INDUV',              INDEX OF DEVPT VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',         INDEX OF SYSTEM NON VSAM FILES
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//*:       VSAMCAT='$VCAT',                    USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                  SYSTEM VSAM CATALOG
//         OUT=$OUT,                                OUTPUT CLASS
//         UTM1FILE=,                         OUTPUT FILE FOR UTM2
//         VOLS='SER=$VOLUN',                       BACKUP VOLUME
//         UNITS=$UNITUN,                             BACKUP UNIT
//         SPAME='(TRK,(300,10),RLSE)'              PC FILE SPACE
//* ------------------------------------------------------
//VERIFY EXEC PGM=IDCAMS
//*---------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//PTUME1  EXEC PGM=BVPTUME1
//*------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7ME   DD DSN=&UTM1FILE,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAME,
//         DCB=(RECFM=FB,BLKSIZE=22100,LRECL=221)
//PAC7EQ   DD SYSOUT=&OUT
```

UTM2 - Input / Processing / Results

A '*' line as input to the second step of the conversion (UTM2) with a user code and its password.

If the user code and password are not indicated, an error message is displayed and the procedure cannot be run.

UTM2 - Description of Steps

Conversion of the meta entities, the relations and their calls: PTUME2

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Input	Development Database data
PAC7AN	&INDUV..&BASE.AN	Input	Development Database index
PAC7ME	&UTM1FILE	Input	List of relations calls in all the entities
PAC7AE	&INDSV..BVPAE	Input	Error messages
PAC7MV	&UTM2MV	Output	Update transactions (length = 170)
PAC7ET		Report	Report
PAC7MB	&&UTM2MB	Input	user input

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	&INDUV..&BASE.AR	Output	Development Database data
PAC7AN	&INDUV.. &BASE.AN	Output	Development Database index
PAC7AY	&INDUV..&BASE.AY	Output	Development Database extension
PAC7AJ	DUMMY		Development Database Journal
PAC7AE	&INDSV..BVPAE	Input	Error messages
PACGGN	&INDSV..BVPGN	Input	Administration Database index
PACGGR	&INDSV..BVPGR	Input	Administration Database data
PACGGY	&INDSV..BVPGY	Input	Administration Database extension
PACGGU	&INDSV..BVPGU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS elements file of the Development Database
PAC7ME	DUMMY		Work file

Code	Physical name	Type	Label
PAC7MV	&UTM2MV	Input	Update transactions
PAC7RB	DUMMY		UPDT erroneous transactions (length=80)
PAC7RY	DUMMY		UPDP erroneous transactions (length=310)
PAC7IE	DUMMY	Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

UTM2 - Execution JCL

```
//* ------------------------------------------------------
//*      VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*            - MIGRATION META-ENTITES OLD 2 -
//*
//* ------------------------------------------------------
//BVPUTM2  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//         INDUV='$INDUV',              INDEX OF DEVPT VSAM FILES
//         INDSV='$INDSV',             INDEX OF SYSTEM VSAM FILES
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//         STEPLIB='$HLQ..SBVPMBR8',      LIBRARY OF LOAD-MODULES
//         LSR='BLSR',                      LSR BATCH SYSTEM NAME
//*:       VSAMCAT='$VCAT',                     USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',                   SYSTEM VSAM CATALOG
//         OUT=$OUT,                                 OUTPUT CLASS
//         UTM1FILE=,                          INPUT FILE FROM UTM1
//         UTM2MV=,                           OUTPUT FILE FROM UTM2
//         VOLS='SER=$VOLUN',                         BACKUP VOLUME
//         UNITS=$UNITUN,                               BACKUP UNIT
//         UWK=$UWK,                                      WORK UNIT
//         SPAMV='(TRK,(300,10),RLSE)'              PC FILE SPACE
//* ------------------------------------------------------
//INPUT  EXEC PGM=BVPTU001
//*----------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
```

```
//PAC7MB   DD DSN=&&UTM2MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//PTUME2  EXEC PGM=BVPTUME2
//*------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//PAC7MB   DD DSN=&&UTM2MB,DISP=(OLD,PASS)
//PAC7ME   DD DSN=&UTM1FILE,DISP=SHR
//PAC7MV   DD DSN=&UTM2MV,DISP=(,CATLG),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAMV,
//         DCB=(RECFM=FB,BLKSIZE=17000,LRECL=170)
//PAC7ET   DD SYSOUT=&OUT
//PACA15 EXEC PGM=BVPACA15,COND=(0,NE,PTUME2)
//*---------------------------------------
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSUDUMP DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//GNLSR    DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGN   DD SUBSYS=(&LSR,'DDNAME=GNLSR','BUFND=10','BUFNI=10')
//GRLSR    DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGR   DD SUBSYS=(&LSR,'DDNAME=GRLSR','BUFND=10')
//GYLSR    DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGY   DD SUBSYS=(&LSR,'DDNAME=GYLSR','BUFND=10')
```

```
//GULSR    DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGU   DD SUBSYS=(&LSR,'DDNAME=GULSR','BUFND=10','BUFNI=10')
//PAC7AJ   DD DUMMY
//ANLSR    DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AN   DD SUBSYS=(&LSR,'DDNAME=ANLSR','BUFND=40','BUFNI=30')
//ARLSR    DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AR   DD SUBSYS=(&LSR,'DDNAME=ARLSR','BUFND=40')
//AYLSR    DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7AY   DD SUBSYS=(&LSR,'DDNAME=AYLSR','BUFND=10')
//PAC7DC   DD DUMMY
//PAC7ME   DD DUMMY
//PAC7MV   DD DSN=&UTM2MV,DISP=SHR
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//PAC7IE   DD DUMMY
//PAC7IF   DD SYSOUT=&OUT
```

SMP/E : Delete a prior version

Introduction

This utility (not provided) is used to delete a VA Pacbase version in SMP/E.

The PDS of 'Target zones' (hlq.SBVPxxx) and 'Distribution zones' (hlq.ABVPxxx) files are purged from the components of the previous version and are available for the new installation.

Implementation:

- Define an MCS (see below), replacing HBVPxxx with the FMID of the version to be deleted.
- Create and execute the JCL presented below, after valorizing its parameters. This JCL calls the MCS defined just before.

Examples

```
MCS TO BE DEFINED :
-----------------

++FUNCTION(DELFUNC)
DESCRIPTION(VISUALAGE PACBASE - CICS)
/*    (C) COPYRIGHT IBM CORP 1983, 2003  */.
++VER(Z038)
     DELETE(HBVPXXX) .


JCL TO BE EXECUTED :
------------------

//$$$$DDEF JOB ($$$$$$),'DDEF',CLASS=
//*-------------------------------------------------------------
//*   PERFORM SMP/E DELETING SYSMOD
//*
//*   BEFORE USING THIS JOB STEP, YOU WILL HAVE TO MAKE THE
```

```
//*   FOLLOWING MODIFICATIONS.
//*
//* - CHANGE THE JOB CARD TO MEET THE INSTALLATION REQUIREMENT
//* - CHANGE &SMPE   BY THE NAME OF SMPE HLQ ON YOUR SYSTEM
//* - CHANGE &TGT    BY THE NAME YOU CHOOSE FOR TARGET ZONE
//* - CHANGE &DLIB   BY THE NAME YOU CHOOSE FOR DISTRIBUTION ZONE
//* - CHANGE &USER   BY THE PREFIX YOU CHOOSE FOR MCS
//* - CHANGE HBVPXXX BY THE FMID YOU WANT TO DELETE
//*
//*-------------------------------------------------------------
//COPT    EXEC PGM=GIMSMP,COND=(4000,LT),REGION=0M
//SMPCSI   DD   DSN=&SMPE.GLOBAL.CSI,DISP=OLD
//SMPPTFIN DD DSN=&USER.SMPMCS,DISP=SHR
//SMPHOLD   DD DUMMY
//SYSPRINT  DD SYSOUT=*
//SMPCNTL DD *
   SET BDY(GLOBAL)           /* SET TO GLOBAL ZONE.       */.
     RECEIVE S(DELFUNC)      /* RECEIVE THE FUNCTION.     */.
   SET BDY(&TGT)             /* SET TO APPLICABLE TARGET. */.
     APPLY S(DELFUNC)        /* APPLY TO DELETE OLD FCT.  */.
   SET BDY(&DLIB)            /* SET TO APPLICABLE DLIB.   */.
     ACCEPT S(DELFUNC)       /* ACCEPT TO DELETE OLD      */.
   SET BDY(&TGT)             /* SET TO APPLICABLE TARGET. */.
   UCLIN.
     DEL SYSMOD(DELFUNC)     /* DELETE SYSMOD ENTRIES FOR */.
     DEL SYSMOD(HBVPXXX)     /* DUMMY AND OLD FUNCTION.   */.
   ENDUCL.
     SET BDY(&DLIB)          /* SET TO APPLICABLE DLIB.   */.
   UCLIN.
     DEL SYSMOD(DELFUNC)     /* DELETE SYSMOD ENTRIES FOR */.
     DEL SYSMOD(HBVPXXX)     /* DUMMY AND OLD FUNCTION.   */.
   ENDUCL.
   SET BDY(GLOBAL)           /* SET TO GLOBAL ZONE.       */.
     REJECT HOLDDATA NOFMID  /* REJECT SYSMODS, HOLDDATA  */
     DELETEFMID              /* FOR THE DELETED FUNCTIONS.*/
     (DELFUNC HBVPXXX)       /* DELETE THE FMIDS FROM THE */
                             /* GLOBALZONE ENTRY.         */.
```

Efficiency enhancement

The performance of the VisualAge Pacbase system may be improved in batch or on-line mode in several ways, depending on the environment and on the size of the managed VA Pac Databases.

This chapter provides information necessary for efficiency enhancement.

The file, program and JCL installation default options may be modified according to the suggestions described below. It is also possible to enhance efficiency by modifying the system environment, which is not described in this chapter since it depends on the specifications of each site.

Moreover, it is recommended not to modify too many chosen default options, so that the VA Pac Database Manager can retrieve these modifications easily when reinstalling the system.

Installation of the Database files

Where possible, the AN, AR and AY files should be copied onto separate volumes.

For large VA Pac Databases (more than one million data items), it may also be appropriate to allocate the files onto several volumes.

Since the AN file is used in its entirety by the system, it is possible to carry out a multi-volume installation simply by modifying the file's DEFINE in the DFbaseAN member of the VA Pac Database parameter PDS.

Resources required under CICS

In order to improve the VA Pac system efficiency under CICS, two objectives should be pursued:

- reduce program loading time,
- make VSAM buffer management more efficient.

Program management

The first solution is fairly easy to implement: it requires a frequency survey on program runs. As a result, the programs that are most often used are said to be resident in the CICS partition (RESIDENT(YES) in CSD).

For example, here are the programs most commonly used by the system:

- BVPRADM : Administration Database access subprogram,
- BVPFCHK : read/write of PUF communication area,
- BVPF000 : distributor for PUF folder,
- BVPFCTL : authorizations control,
- BVPFANM : VA Pacbase choice,
- BVPSERR : errors server,
- BVPSFF0 : meta entities folder,
- BVPR980 : 3270 messages formatting,
- BVPSFIN : meta entities folder initialization,
- BVPR000 : COMM.3270,PUF-TP,V.LOG server.

Buffer management

Three parameters associated with the files must be entered in CSD: STRINGS, DATABUFFERS, and INDEXBUFFERS.

- STRINGS:

  It specifies the number of strings or requests which can be executed n parallel by VSAM on the same file: usually, a string is used during an input/output on disk.

  Certain commands, however, do not follow this rule:

  READ UPDATE and STARTBR or RESETBR reserve a string up to the REWRITE or UNLOCK and ENDBR commands, respectively.

- DATABUFFERS:

  It specifies the number of DATA buffers to be allocated by VSAM upon the file opening. A minimum of two buffers is required: one for the reading of the 'control interval DATA', and one for the 'split' of the 'control interval'.

  Default: DATABUFFERS = STRINGS + 1

- INDEXBUFFERS:

  It specifies the number of INDEX buffers to be allocated by VSAM upon the file opening. The minimum number of buffers per request is one.

  It is preferable, however, to increase this number because for a file with three INDEX levels, three EXCP's are required to point to the searched CI DATA for each on-line request.

  It is recommended to allocate:

  – at least: one buffer per index level, plus one buffer per string (STRING value).

  – no more than: one buffer per CI of INDEX SET, plus one buffer per string.

Conclusion

A sufficient number of buffers should be provided for, according to the number of strings.

However, too many buffers may lead to an excessive pagination, and therefore EXCP's.

Part Number: DEPCI001355A - 7324

Printed in USA