VisualAge Pacbase

Installation Guide
IMS/VS Server & Client components

Version 3.5

VisualAge Pacbase

Installation Guide
IMS/VS 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.

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 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'.

- 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: IMS DB/DC

Installation media: DAT reader 4 mm or cartridge 3480

Monitor: IMS/DC Version 6

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'.

Communication

To enable the communication between the workstation components and the servers in an IMS/VS environment, the possible communication protocols are:
- MVS IMS Connect,
- MQ-IMS Bridge.

Chapter 3. Installation of Server Components

Environment

Introduction

This chapter introduces the environment and resources required to be able to use VisualAge Pacbase.

TP Environment

The Monitor used for TP VisualAge Pacbase is IMS/DC.

Since Version 2.5 of VA Pac, the MFS system is bypassed by the use of the DFS.EDTN standard module for the receiving and display of messages.

The formatting of messages is managed by the BVPR980 sub-program found in the on-line programs library.

Only one transaction code is to be declared in IMS to access VisualAge Pacbase:

- tttt transaction code to connect to VisualAge Pacbase - on-line mode (Input Point),

  This VA Pac online transaction, to be declared in IMS, is transactional and uses a SPA with a length of 150 for VisualAge Pacbase.

DL/1 databases organization

VisualAge Pacbase is its own Database Manager. It only uses DL/1 to store records that support the physical organization of the database.

Consequently:

- The user cannot directly access VisualAge Pacbase data using DL/1 utilities but must use the tools designed and supplied for this purpose.
- The DL/1 facilities needed by VisualAge Pacbase are greatly reduced. Specifically, all of the databases are built from a single root segment. Thus, there are no dependent segments and the DBRs have fixed lengths.
- Secondary indexes and logical relationships are never used. This results in a PHYSICAL ACCESS to DL/1 ACCESS ratio that is considerably less than the ratios that are generally found in current applications.

DL/1 databases organization

Three types of DL/1 organization are used: HDAM, HISAM, and HIDAM.

HDAM-OSAM VSAM databases

This organization is used for the Journal (AJ GJ) and Data (AR GR) databases, for the extension (AY GY) and for the traces of PUF communication areas (TR). It is also used for the archival journal of VA Pac interface - configuration management (QJ).

The key to these files is a eight-byte numeric field in ascending order without a break in sequence. This record number is a logical pointer that is internal to the system.

Each record is attached to an anchor point in a one-to-one relationship (Anchor Point DL/1) calculated by the randomization module CGIPACR1. The assembler source of this module is supplied with VisualAge Pacbase in the SY Parameters file under the name of 'RANDOM'. It must be compiled in the site's IMS RESLIB. This module is directly derived from the randomization module modulo DFSHDC10. The only difference is correction of the CI number; this is done in order to avoid addressing in the 'BIT MAPS'.

Placement in the physical file is illustrated in the following example (one block may contain 25 DBR):

VA Pac key	CI Number	AP Number
1	2	1
2	2	2
3	2	3
...	...	...
...	...	...
25	2	25
26	3	1
27	3	2
...	...	...
etc...		

The results are as follows:
- A synonym chain never appears, thus no FREE SPACE or OVERFLOW AREA is to be anticipated.

- The physical file is used according to the ascending RBA's. Therefore, the OSAM space can be allocated according to the real volume of the database, independent of the number of CI's which could be addressed by the randomization module.
- The description of the RMNAME macro is: RMNAME=(CGIPACR1,X,Y) where:
  - X = the number of ANCHOR POINTs per CI. X is calculated according to the size of the VisualAge Pacbase record and the size of the CI using the following formula: $X = INT((CI - 19) / (REC + 11))$ where:
    - INT = integer function.
    - CI = size of the CI in bytes.
    - REC = size of the VisualAge Pacbase record in bytes.
  - Y = number of CI's which could be addressed. Keeping in mind that the CI is an incremental function of the VisualAge Pacbase key and that this key is assigned in ascending sequence, the user may choose a high value for the number of CI's (limited to ((2 exponent 24) - 1) without affecting physical organization or performance, thus avoiding possible overflows.

    It may be interesting to choose a very high value to avoid any possible overflow, 16,000 000 for example.

    EXAMPLE: Let's consider the (AR) database:

    Size of the VisualAge Pacbase record : 152

    Size of the CI in use : 4,096

    ---> X=26

    ---> RMNAME=(CGIPACR1,26,16000000)

HISAM VSAM Databases

This organization is reserved for the Index database (AN, GN), Documentation (AE), Review of Jobs (LB), SPA database (SV) user database (GU) and PUF working database (WS).

The physical DL/1 record contains one and only one complete DBR, which means one and only one VisualAge Pacbase record plus DL/1 control information. There is no DATASET OVERFLOW.

During loading, the size of the database is calculated directly from the number of logical records, the RECORD size, and the amount of FREE SPACE requested at the time of the DEFINE of the VSAM file.

HIDAM VSAM databases

This organization is reserved for the VisualAge Pacbase components database DC whose first index is made up by the D3 database and for the P.A.F. working databases PA whose first index is made up by the P1 database.

Each DBR for the DC and PA databases includes only one variable length root segment.

Optimization of performance

Installation of the index (AN) and data (AR) files

It is recommended that the AN and AR files be installed on separate volumes, specifically when the VA Pac Database stores more than one million data.

Knowing that the whole of the AN file is accessed by all VisualAge Pacbase operations, its DEFINE may be adapted in the DF$BASE.AN member found in the database parameters PDS so as to provide for a multi-volume installation. For more information, please refer to chapter 'Installation of the Server Environment'.

Regarding the Data (AR) file, in most cases, 80 per cent of VisualAge Pacbase operations access the data stored at the BEGINNING of the file. Therefore, simply storing the AR file on several volumes cannot improve access performance.

Thus, a standard procedure, named STOP for STorage OPtimization, makes a multi-volume installation effective by outspreading the most frequently accessed data throughout the whole of the file.

The definition on several volumes of the AR file combined with the execution of the STOP procedure must be done before the Database Restoration.

For complete information on the STOP procedure, refer to the - Batch Procedures - The Administrator's Guide, chapter 'Manager's Utilities', subchapter 'STOP - Storage Optimization of Multi-Volumes'.

Environment Preparation

During this phase, the system must be prepared for the installation of the VisualAge Pacbase system, taking into account its technical characteristics which are described in the first two chapters of this manual:

- Choose the suffixes which are to be assigned to the various VisualAge Pacbase files and codes,
- Choose the VSAM catalogs and allocate adequate disk space,
- And, in particular, prepare the IMS/DC generation, assigning the appropriate values to the parameters indicated below.

Parameters for IMS/DC generation

In the parameters listed below, the yyyy symbol represents the database code used as the suffix of the DBDs. The zz symbol represents the suffix of the batch PSBs. These symbols are chosen by the user.

The installation is simpler if the following values are used, in which case, the compilation of DBDs and PSBs is not necessary:
- zz = 35 for the suffix of batch PSBs
- yyyy = P350 for the suffix of the DBDs

Declaration of the DBDs to be used:

DBD name	DATASETS names
standard...	
BVPDAE	BVP7AE
BVPDSV	BVP7SV
BVPDLB	BVP7LB
BVPDGN	BVP7GN
BVPDGR	BVP7GR
BVPDGU	BVP7GU
BVPDGY	BVP7GY
BVPDGJ	BVP7GJ
BVPDQJ	BVP7QJ
On-line....	
TDANyyyy	T7ANyyyy
TDARyyyy	T7ARyyyy
TDAYyyyy	T7AYyyyy
TDAJyyyy	T7AJyyyy
TDDCyyyy	T7DCyyyy
TDD3yyyy	T7D3yyyy
BVPDPA	BVP7PA
BVPDP1	BVP7P1
BVPDWS	BVP7WS
BVPDTR	BVP7TR
Batch......	
BDANyyyy	B7ANyyyy
BDARyyyy	B7ARyyyy
BDAYyyyy	B7AYyyyy

DBD name	DATASETS names
BDAJyyyy	B7AJyyyy
BDDCyyyy	B7DCyyyy
BDD3yyyy	B7D3yyyy
BDPAyyyy	B7PAyyyy
BDP1yyyy	B7P1yyyy
Admin with PAC Segment	
BVPAGN	BVP8GN
BVPAGR	BVP8GR
BVPAGY	BVP8GY
BVPAGJ	BVP8GJ

DATABASE ACCESS=UP,DBD=(BVPDAE,BVPDSV,etc ...)

Declaration of the VisualAge Pacbase transaction codes:

```
APPLCTN  PSB=BVPRIMS
TRANSACT CODE=tttt,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,
         PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
         MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150)
```

where tttt represents the transaction code chosen for the VisualAge Pacbase connection ($BASE parameter).

Note: The use of RESPONSE mode for VisualAge Pacbase transactions is recommended.

Declaration of the BMP

```
   APPLCTN  PSB=PACBzz   (VA PAC output generation)
```

Note: Parameter sources for IMS generation are supplied on the tape in the hlq.SBVPINST.

Parameterization

Installation Preparation

The preparation to the installation process consists of three phases:

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

This stage must be performed by the system staff who are accustomed to installing IBM products with SMP/E.

- Allocation of a PDS file meant to save all the installation and operation JCL.
- Execution of the installation JCL from the PDS members created during the first phase.

Remark for a previous SMP/E installation

For the following phase, the SMP/E context is supposed to be virgin for the product. Either the product has never been installed with SMP/E, or the different files relating to SMP/E have been reinitialized for the previous version, or a new SMP/E environment will be created for this version.

In the contrary, to reuse the SMP/E files of the previous version, it is possible to execute, before the first phase, the SMP/E utility described in the appendix of this manual.

First phase

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

This stage consists of the following steps:
- Downloading of JCL examples via IEBCOPY
- Setting up of the SMP/E environment if required
  - Definition of SMP/E cluster libraries (BVP1DCSI)
  - Initialization of SMP/E libraries (BVP2ICSI)
  - Allocation of SMP/E work files (BVP3ALLO)
  - FMID assignment (BVP4DEFZ)
  - Input creation: DDDEF technical files (BVP5DDEF)
- Cartridge receipt
  - Definition of 'Target field' and 'distribution field' (BVP6DDEF) files
  - Assignment of files 'Target field' and 'Distribution field' (BVP7ALLO)
  - Execution of RECEIVE (BVPRECV)
  - Execution of APPLY (BVPAPPLY)
  - Execution of ACCEPT (BVPACCPT)

Once this phase has been executed, all components required at installation are to be found in the following PDS, 'hlq' indicating the common prefix of delivered elements (High-Level Qualifier):
- hlq.SBVPINST: all the files with a length of 80 including:
  - The error messages file,
  - The initial installation JCL,

- A file including the installation and operation JCL's and procedures (BVPTOTAL),
- hlq.SBVPF2: PC and PE initialization files,
- hlq.SBVPMBR8: batch load-modules (BVPDS600 and BVPDS610 interface load-modules between VA Pacbase and DSMS 2.5 are provided in standard; for a DSMS use, the DEXP procedure must be adapted to use these new load-modules in place of PDS600 and PDS610 which are not compatible since VA Pacbase 3.0).
- hlq.SBVPMTR8: on-line 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 file
- hlq.SBVPDBD: dbd
- hlq.SBVPPSB: psb

Second phase

This phase is optional but it is recommended to execute it. It involves allocating a PDS file with the following characteristics:
- Lrecl=80
- Size: around 100 disc track 3,390, 30 pads directory.

Third phase

The third phase involves copying the 'hlq.SBVPINST(BVPINIT)' JCL into the PDS mentioned in the second phase, modifying its parameters to match the constraints of the site and executing it so as to obtain a complete installation and operating JCL.

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

It must be completed as follows:
- Fill in '&hlq' with the 'hlq' parameter value used in the first phase SMP/E.
- 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 JCL's, 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 delivered with their password in the provided Administration Database and are used in the delivered 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(BVPTOTAL),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 CHARACTER
===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 INDSN='EXP.BVP3N'  .SYSTEM NON VSAM FILES INDEX VA-PAC
===PRM INDUV='UTI.BVP3V'  .USER VSAM FILES INDEX
===PRM INDUN='UTI.BVP3N'  .USER NON VSAM FILES INDEX VA-PAC
===PRM BASE='P350'        .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 UNITSV=SYSDA       .VSAM SYSTEM FILES UNIT
===PRM UNITUN=SYSDA       .NON VSAM USER FILES UNIT
```

```
===PRM UNITUV=SYSDA        .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 BIBP='SYS1.PROCLIB'  .PROCEDURE LIBRARY
===PRM DSCB='BVP.DSCB',     .DSCB MODEL FILE
===PRM HLQ='HLQ'            .HIGH LEVEL QUALIFIER OF LOAD MODULES
===PRM BIBT='SYS1.SORTLIB'    .SORT LIBRARY
===PRM BCOB='SYS1.SCEERUN'    .COBOL ROUTINE LIBRARY
===PRM TABTDF=<>             .DSN OF TABLES DESCRIPTION FILE
===PRM DSMS=<>              .INDEX OF PRODUCT ELEMENTS (DC)
===PRM CSOC='DBSS'          .TRANSACTION CODE ADWORKBENCH
===PRM REGSIZ='1536K'        .SIZE OF THE REGION FOR BATCH PROCED
===PRM IMSID='IM16'          .'IMSID' PARAMETER FOR BMP PROCEDURE
===PRM IRLM=N               .USING IRLM IN BATCH PROCEDURES (N=N
===PRM DBRC=N               .USING DBRC IN BATCH PROCEDURES (N=N
===PRM SUG='35'             .SUFFIX OF VA PAC'S PSB
===PRM DBDLIB='&HLQ..SBVPDBD'  .DBD LIBRARY
===PRM PSBLIB='&HLQ..SBVPPSB'  .PSB LIBRARY
===PRM ACBLIB='&HLQ..ACBLIB'  .ACB LIBRARY
===PRM RESLIB='IMSVS.RESLIB'  .IMS RESLIB
===PRM PRCLIB='IMSVS.PROCLIB'  .IMS PROCLIB
===PRM CLS='2'              .CLASS FOR TRANSACTION CODES
===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 skeleton JCL, which are called 'JCL modules',
- Parameterize the skeleton in order to obtain a JCL requiring a minimum of modifications to make it operational,
- Add lines before/or after the JCL modules to separate them.

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
===BEGMOD		Insertion of lines at the beginning of module

Command	Parameters	Comments
....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 period and it should not exceed column 72.

The default values are only examples. You must therefore enter values matching with the choices on 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 resulting in SYSUT2 contains all the installation and operation JCL. This file must be run in 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 are entered as comments in the installation JCL:
- In the DELETE/DEFINE*/
- In the JCL STEPCAT's
- In the procedure 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, affected lines should be changed into command lines. This is accomplished by:
- Transforming all '//*:' into '//',
- Substituting blanks for '/*:' and '*/'.

Caution: SMS
- If the SMS product is installed, you should delete IDCAMS definition DD //GDGMOD lines in the installation JCLs with GenerationDataGroup allocation.
- If the UNIT and VOL parameters cannot be used on the site, you can delete them in the whole JCL through an exclusion (EXCLUDE command of TSO/EDIT).

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

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).

This allows for the complete JCL to be split in as many members as there are JCL modules in a PDS. The completed 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.

<u>Report</u>

BVPMMJCL produces a list for each JCL module created, including parameters taken into account and according to required variants.

Note:

Since the JCL skeleton parameters are in the $xxxx format, during execution, if BVPMMJCL encounters a $ character that 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 the '$' character in the flow processed by BVPMMJCL and which is NOT a parameter.

JCL List

<u>Table of installation JCLs</u>

Member	Contents	Procedure
	System and Administration Database installation	
D01ALLOC	Allocation of parameter PDS's	
D02CPAR	Loading of common parameters	
D03DEFIN	Common file definition	
D03GDG	Definition of GDG for parameter backup	
D03INI	Initialization of system files	
D04MBR	Renaming of Exit users batch load modules	
D04PSREN	Naming of PSBs TP	
D04SRCD	Loading of DBDs sources	
D04SRCP	Loading of PSBs sources	
D04ACB	Compiling ACBs	
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
D07INIT	Initialization of working libraries	LDLB LDSV LDWS LDP1
D08INGU	Initialization of users code file	INGU
D08INIAD	GDG definition and initialization	
D08RSAD	Backup of Administration Database	RSAD

Member	Contents	Procedure
D08TINQJ	Initialization of archival journal file VA Pacbase interface	INQJ
D08XMET	Installation of administration Model	VING
D99INSL	List of components' dates	INSL
	Installation of Development Database	
I01SY	Allocation of parameter PDS's	
I02SY	Loading of parameters	
I03DEF	Definition of development databases	
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	Archival of the Database journal	ARAD
JCLPAGX	Extraction of Administration Database	PAGX
JCLROAD	Reorganization of Administration Database	ROAD
JCLRSAD	Backup 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	Archival 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

Member	Contents	Procedure
JCLEXLI	Extraction of libraries	PACX
JCLEXPJ	Extraction of transactions from journal	PACX
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
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	Update of batch	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	
JCLUPPM	Update of development base	UPPM
JCLARPM	Archival of SCM journal (QJ)	ARPM
JCLCHPM	Integrity Control of Events/Elements	CHPM

Member	Contents	Procedure
JCLCPPM	Comparison with extracted files	CPPM
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 inputs	ISEP
JCLISOS	Selection of strings, operators	ISOS
	Retrieval of former releases	
RGY250	Exchange of GY transactions from the 2.n	GY25
RMB250	Exchange of MB transactions from the 2.n	MB25
RGY300	Exchange of GY transactions to the 2.n	GY30
RMB300	Exchange of MB transactions to the 2.n	MB30
RMP250	Retrieval of passwords from the 2.5	UTMP

Member	Contents	Procedure
RPC250	Retrieval of PC file from the 2.n	PC25
RPE250	Retrieval of PE file from the 2.n	PE25
RPG200	Retrieval of PG file from the 2.0	PG20
RPG250	Retrieval of PG file from the 2.5	PG25
RPJ250	Retrieval of PJ file from the 2.n	PG25
RPP250	PEI retrieval from 2.n on	PP25
RUV250	Retrieval of UV file from Pac/Transfer	UV25
	UTILITIES	
JCLSTAT	Generation statistics file on base	STAT
	'P' lines extraction with 'UNS' operator	UTU1
	'P' lines update with 'UNS' operator	UTU2

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 column 72

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

Note:

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

Parameters table

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'

Parameter	Meaning	Default
INDSN	VA Pac Non-VSAM system (SAM,PDS) (maximum: 24 characters)	'EXP.BVP3N'
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)	P350
OUT	SYSOUT print class	H
UWK	UNIT of work files used	SYSDA
UNITSN	UNIT of NON-VSAM system files	SYSDA
UNITSV	UNIT of VSAM system files	SYSDA
UNITUN	UNIT of NON-VSAM user files	SYSDA
UNITUV	UNIT of 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	A
DSCB	DSNAME of DSCB model file for generation files	'BVP.DSCB'
BIBP	DSNAME of the procedure library	'SYS1.PROCLIB'
HLQ	Prefix of the batch and on-line load-module library (30 char. max)	'HLQ'
BIBT	DSNAME of sort library (SORTLIB)	'SYS1.SORTLIB'
BCOB	DSNAME of COBOL routine library	'SYS1.SCEERUN'
	Pactables	
TABTDF	Table description file DSN (1)	<>
	DSMS module	
DSMS	Development database-element file name (1)	<>
CSOC	AD workbench Transaction code	DBSS
REGSIZ	Region size for batch procedures	1536K

Parameter	Meaning	Default
IMSID	IMSID parameter for batch procedures (N=NO)	N
IRLM	Use of ILRM in batch procedures (N=NO)	N
DBRC	Use of DBRC in batch procedures (N=NO)	N
SUG	VA Pac PSBs suffix (2 characters)	35
DBDLIB	VA Pac DBDs library (2)	$HLQ.DBDLIB
PSBLIB	VA Pac PSBs library (2)	$HLQ.PSBLIB
ACBLIB	ACB library	$HLQ.ACBLIB
RESLIB	IMS RESLIB	IMSVS.RESLIB
PRCLIB	PROCLIB IMS	IMSVS.PROCLIB
CLS	Transaction codes classes	2

(1) The files DSN should be replaced by those that are installed on site only if the Tables or DSMS modules are installed or if the default name does not fit.

(2) Careful, if the chosen suffix ($SUG parameter) is set to 35 and the database code ($BASE parameter) is set to P350, the $DBDLIB parameter prefix must have the value of the $HLQ parameter followed by SBVPDBD and the $PSBLIB parameter must have the value of the $HLQ parameter followed by SBVPPSB.

Example : if the $HLQ parameter is set to EXP.PB350, the $DBDLIB parameter will have the value EXP.PB350.SBVPDBD and the $PSBLIB parameter will have the value EXP.PB350.SBVPPSB.

JCL Before/After Lines

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 in the 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

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 : DELETE                                            *
//*      .STEP2 : ALLOCATION                                        *
//********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..BVPSY)
//*
//STEP2  EXEC PGM=IEFBR14
//SY       DD DSN=$INDSN..BVPSY,DISP=(,CATLG,DELETE),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6080),
//         VOL=SER=$VOLSN,
//         UNIT=$UNITSN,
//         SPACE=(6080,(50,,10))
//
```

D02CPAR module: '$prfj.SY' 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

```
//$PRFJ.SY JOB ($CCPT),'PAC D02CPAR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//********************************************************************
//* VISUALAGE PACBASE                                               *
//*                                                                 *
//* INSTALLATION  : LOADING PARAMETERS IN PDS                       *
//*                                                                 *
//*---------------------------------------------------------------  *
//*                 I M P O R T A N T                               *
//*                 -----------                                     *
//*      BEFORE EXECUTING THIS JOB, REPLACE ALL                     *
//*          ':/'  BY './' UNDER THE EDITOR.                        *
//*---------------------------------------------------------------  *
//*                                                                 *
//********************************************************************
//*
//STEP1    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=$OUT
//SYSUT2   DD DSN=$INDSN..BVPSY,DISP=SHR
//SYSIN    DD DATA,DLM='PP'
:/ ADD NAME=RANDOM
HDC1     TITLE 'HDAM RANDOMIZING MODULE IMS/VS'
CGIPACR1 CSECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                            *
*          S A M P L E   C O N V E R S I O N   P R O G R A M *
*                                                            *
*     THIS CSECT CONVERTS AN EBCDIC NUMERIC KEY TO A RELATIVE *
* BLOCK AND ROOT ANCHOR POINT. THIS RESULT IS OBTAINED AS     *
* FOLLOWS  RECNO= MOD(KEY,DMBDABLK*DMBDARAP)                  *
*          BLOCK= RECNO/DMBDARAP+NB DE BIT MAP AVANT          *
*          RAP  = MOD(RECNO,DMBDARAP)+1                       *
*     THE CSECT ASSUMES THAT THE EXTERNAL KEY IS 15 BYTES OR  *
* LESS.  NON-NUMERIC CHARACTERS ARE VALID, HOWEVER ONLY THE   *
* FOUR LOW ORDER BITS WILL BE USED.                          *
*                                                            *
* CALLING SEQUENCE                                           *
*      R0 - DMB                                              *
*      R1 - DMBDACS                                          *
*      R7 - PST                                              *
*      R9 - KEY ADDRESS                                      *
* ON RETURN                                                  *
```

```
*         DMBDACP - BBBR                                          *
*---------------------------------------------------------------*
*---------------------------------------------------------------*
*    THE CGIPACR1 RANDOMIZATION MODULE IS DERIVED FROM THE       *
* MODULO RANDOMIZATION DFSHDC10 SUPPLIED BY IBM.                 *
*    THE CHARACTERISTICS ARE THE FOLLOWING :                     *
*      - BLOCK=RECNO/DMBDARAP+1(+ N) IN ORDER TO AVOID ADDRES-*
* SING THE NTH 'BIT MAP'.                                        *
*      - PROTECTION WHEN RECEIVING A NON NUMERIC KEY.            *
* IN SUCH CASE, THE SET ROOT ANCHOR POINT IS FOUND IN THE        *
* 'BIT MAP' (INSTEAD OF A S0C7 ABEND IN THE CONTROL REGION.      *
*         NOTE FOR DFSHDC10 :                                    *
*  THE USE OF THE DFSHDC10 MODULE IS TRANSPARENT FOR ALL         *
* PACBASE PROCESSING AS IT DOES NOT INVOLVE ANY MODIFICATION     *
* IN THE PHYSICAL SEQUENCE OF ROOTS. HOWEVER, A SYSTEM DEGRA-    *
* DATION OCCURS : HDAM DATABASE I/O IS TWICE AS LONG AND THE     *
* CONTROL REGION IS SENSITIVE TO PROGRAM ANOMALIES.             *
*                                                               *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          STM   14,12,12(13)          SAVE
          USING CGIPACR1,R15
          B     EPA1
          DC    CL20'&SYSDATE &SYSTIME'
EPA1      DS    0H
          USING DMB,R10
          LR    R10,R0
          LR    R11,R7
          USING WRKAREA,R11
          USING PST,R7
          USING DMBDACS,R1
          XC    TMPDECB(9),TMPDECB    INIT FOR CVB
          IC    R5,DMBDAKL            GET EXECUTABLE KEY FLD LENGTH
          EX    R5,MVC                TRANSFER KEY (R9) IN BUFFER AMPON
          LR    R4,R1                 SAVE DMBDACS ADDRESS BEFORE RT
          TRT   TMPDECB(1),TBALPHA    IS THE 1ST CARACTER ALPHA ? A ?
          BNZ   SUITE                 NO  : BRANCHING
          MVC   PSTDECB(7),TMPDECB    YES : BUFFER FIELD RIGHT    MPON
          MVC   TMPDECB+1(7),PSTDECB        SHIFT OF ONE BYTE      ITE
          TR    TMPDECB(1),TBNUM1     OVERRIDE 1ST BYTE ALPHA ->   NUM
          TR    TMPDECB+1(1),TBNUM2   OVERRIDE 2ND BYTE ALPHA ->   NUM
          LA    R5,1(R5)              THE KEY IS 1 BYTE LONGER     PLUS
SUITE     LR    R1,R4                 RESTORATION DMBDACS ADDRESS
          EX    R5,TRT                IS THE KEY NUMERIC ?
          BNZ   NONUM                 NO  : BRANCHING
          XC    PSTDECB(8),PSTDECB    INIT FOR CVB
          EX    R5,PACK               PACKING BUFFER FIELD IN PSTDTDECB
          SR    R4,R4
          OI    PSTDECB+7,X'0F'       FORCE SIGN
          SR    R8,R8

COMPARE   EQU   *
          CP    PSTDECB(8),MAXP(6)    IS NUMBER TOO LARGE FOR CVB
          BH    DECR                  YES, BRANCH
          CVB   R5,PSTDECB
          B     ALMOST                FINISH UP
```

```
DECR      EQU     *
          SP      PSTDECB(8),MAXP(6)      DECR NUMBER BY 2147483647
          AL      R8,MAXB                 INCR REG 8 BY SAME AMOUNT
          BC      CARRY,CARRY1            BR IF CARRY OUT OF REG
          B       COMPARE                 OTHERWISE COMPARE AGAIN
CARRY1    EQU     *
          LA      R4,1(,R4)               TAKE CARE OF CARRY
          B       COMPARE                 GO COMPARE
ALMOST    EQU     *
          ALR     R5,R8                   PUT IF ALL TOGETHER
          BC      NOCARRY,DONE            IF NO CARRY, WE ARE DONE
          LA      R4,1(,R4)               ELSE, TAKE CARE OF CARRY
*                                         EVEN-ODD PAIR 4,5 HAVE
*                                         CONVERTED NUMBER
DONE      EQU     *
          SR      R4,R4                   REINITIALIZE R4 (SAFETY)
          LH      R6,DMBDARAP             SEGMENT NUMBER LOAD PER CI
          DR      R4,R6                   R4 = RAP N° IN BLOCK
*                                         R5 = CORRESPONDING N° CI
*                                         WITHOUT TAKE CARE OF BIT MAP
          LA      R4,1(R4)                +1 TO NOT BEGIN AT 0
          LR      R3,R5                   MEMO
          LA      R6,DMBAMPOF             OFEST AMPB
          AH      R6,0(R6)                ADRESS AMPB
          USING   DMBAMP,R6
          SR      R8,R8
          ICM     R8,B'0011',DMBPFOBK     CI LENGTH IN BYTES
          LTR     R8,R8                   OSAM ?
          BNZ     OKBLK                   YES
          ICM     R8,B'0011',DMBPFPBK     CI LENGTH IN BYTES
OKBLK     DS      0H
          LR      R6,R8
          DROP    R6
          LH      R2,DMBDARAP             ANCHOR POINTS NUMBER
          LA      R2,1(R2)                +1 POUR FSEAP
          SLL     R2,2                    CTRL BIT MAP BYTES NUMBER
          SR      R6,R2                   USEFUL BIT MAP BYTES NUMBER
          SLL     R6,3                    USEFUL BIT MAP BINARY DIGITS
          BCTR    R6,R0                   CI NUMBER DESCRIBE BY THE BIMAP
*                                         (WITHOUT COUNT THE BIT MAP)
          SR      R2,R2
          DR      R2,R6                   R3 = BIT MAPS NUMBER BEFORE
          AR      R5,R3                   CORRESPONDING CI N° -1.
          LA      R5,2(R5)                CORRESPONDING CI N° (+1 FOR
*                                         THE BIT MAP +1 CHAR -
*                                         THE 1ST CI IS THE N° 1 CI)
          SLL     R5,8                    BBB0 FORMAT
          OR      R4,R5                   BBBS FORMAT
          B       FIN
NONUM     DS      0H
          LR      R1,R4                   DMBDACS ADDRESS RESTORATION
          L       R4,BBBR1
FIN       DS      0H
          ST      R4,DMBDACP              RESULT
          LM      14,12,12(13)            RESTORE
```

```
              BR    R14                      RETURN
*
*
PACK     PACK  PSTDECB(8),TMPDECB(0)
MVC      MVC   TMPDECB(*-*),0(R9)
TRT      TRT   TMPDECB(*-*),TABNUM
         REQUATE
CARRY    EQU   3
NOCARRY  EQU   12
*
MAXP     DC    P'2147483647'
MAXB     DC    F'2147483647'            MAX SIGNED 32-BIT NUMBER
BBBR1    DC    X'00000101'
TBALPHA  DS    0CL256
         DC    193X'FF',9X'00',7X'FF'
         DC    9X'00',8X'FF',8X'00',22X'FF'
TBNUM1   DS    0CL256
         DC    193C'?',9C'1',7C'?'
         DC    C'1',8C'2',8C'?'
         DC    2C'2',6C'3',22C'?'
TBNUM2   DS    0CL256
         DC    193C'?',CL9'012345678',7C'?'
         DC    CL9'901234567',8C'?'
         DC    CL8'89012345',22C'?'
TABNUM   DS    0CL256
         DC    240C'9',20X'0',6C'9'
*
         IDLI  PSTBASE=0,DMBBASE=0
WRKAREA  DSECT
         DS    2F
TMPDECB  DS    6F
         END
:/     ADD NAME=DFBVPAE
 DELETE ($INDSV..BVPAE) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPAE)           -
                 SHR (2,3)       KEYS (12,6)      -
                                 REC (250000)     -
                 RECSZ (88 88) RUS )              -
        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 ($INDSV..BVPGN)           -
                 SHR (2)         KEYS (49,6)      -
                 VOL ($VOLSV)                     -
                 CYL (4 1)                        -
                 RECSZ (68,68) RUS )              -
        INDEX   ( NAME ($INDSV..BVPGN.I)          -
```

```
                         CISZ (4096) )                       -
        DATA    ( NAME ($INDSV..BVPGN.D)               -
                  FSPC (10,5)                          -
        CISZ (4096) )                /*: CATALOG ($VCAT) */
:/      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=DFBVPGU
 DELETE ($INDSV..BVPGU) CLUSTER
   DEFINE CLUSTER ( NAME ($INDSV..BVPGU)              -
                  SHR (2,3)       KEYS (8,6)          -
                  VOL ($VOLSV)                        -
                  CYL (4 1)                           -
                  RECSZ (88,88) RUS )                 -
        INDEX   ( NAME ($INDSV..BVPGU.I)              -
                  CISZ (1024) )                       -
        DATA    ( NAME ($INDSV..BVPGU.D)              -
                  FSPC (2,1)                          -
        CISZ (4096) )                /*: CATALOG ($VCAT) */
:/      ADD NAME=DFBVPWS
 DELETE ($INDSV..BVPWS) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPWS)                -
                  SHR (2,3)       KEYS (37,6)         -
                  VOL ($VOLSV)                        -
                  CYL (3 3)                           -
                  RECSZ (1546,1546) RUS )             -
        INDEX   ( NAME ($INDSV..BVPWS.I)              -
                  CISZ (4096) )                       -
        DATA    ( NAME ($INDSV..BVPWS.D)              -
                  FSPC (50,5)                         -
        CISZ (4096) )                /*: CATALOG ($VCAT) */
:/      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 ($INDSV..BVPSG.D)              -
                  FSPC (10,5) SPEED  CYL (3 1)        -
        CISZ (5120) )   /*: CATALOG ($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 (3 1)    -
            CISZ (5120) )   /*: CATALOG ($SCAT) */
:/     ADD NAME=DFBVPSS
 DELETE ($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 ($SCAT) */
:/       ADD NAME=DFBVPSP
 DELETE ($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 ($SCAT) */
:/       ADD NAME=DFBVPSR
 DELETE ($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 ($SCAT) */
:/       ADD NAME=DFBVPLB
 DELETE ($INDSV..BVPLB) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDSV..BVPLB)           -
                       SHR (2,3)           KEYS (23,6)  -
                       VOL ($VOLSV)                    -
                       CYL (2 1)                       -
                       RECSZ (126,126) RUS )           -
            INDEX   ( NAME ($INDSV..BVPLB.I)           -
                       CISZ (4096) )                   -
            DATA    ( NAME ($INDSV..BVPLB.D)           -
                       FSPC (10,5)                     -
```

```
          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 ($SCAT) */
:/      ADD NAME=DFBVPSV
 DELETE ($INDSV..BVPSV) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..BVPSV)     -
                 SHR  (2,3)   KEYS (15,6)            -
                 VOL ($VOLSV)  CYL (3 3)              -
                 RECSZ (16048,16048) RUS )           -
        INDEX   ( NAME ($INDSV..BVPSV.I)     -
                  CISZ (16384) )                      -
        DATA    ( NAME ($INDSV..BVPSV.D)     -
                  FSPC (50,5)                         -
                  CISZ (16384) ) /*: CATALOG ($VCAT) :*/
:/    ADD NAME=DFSYSEXT
 DELETE ($INDUV..SYSEXT.&USER) CLUSTER
 SET LASTCC = 0
 SET MAXCC = 0
 DEFINE CLUSTER ( NAME ($INDUV..SYSEXT.&USER)     -
                 SHR (2 3)  RUS  KEYS(43 0)        -
                 VOL ($VOLUV)                      -
                 CYL (5 1)                         -
                 RECSZ (254 254) )                 -
        INDEX   ( NAME ($INDUV..SYSEXT.&USER.I)    -
                  CISZ (512) )                     -
        DATA    ( NAME ($INDUV..SYSEXT.&USER.D)    -
                  FSPC (10 5) SPEED                -
        CISZ (4096) )          /*: CATALOG ($VCAT) */
:/    ADD NAME=DLSYSEXT
 DELETE ($INDUV..SYSEXT.&USER) CLUSTER
:/    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=DLSYIANA
 DELETE ($INDUV..SYIANA.&USER) CLUSTER
```

```
 :/     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=DLSYTRDU
 DELETE ($INDUV..SYTRDU.&USER) CLUSTER
 :/     ADD NAME=DFSYTRPF
 DELETE ($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 ($VCAT) */
 :/     ADD NAME=DLSYTRPF
 DELETE ($INDUV..SYTRPF.&USER) CLUSTER
 :/     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 (512) )                    -
        DATA    ( NAME ($INDUV..WK&USER.D)   -
                  FSPC (10 5) SPEED               -
        CISZ (4096) )          /*: CATALOG ($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 (1 1)    -
                  RECSZ (1028 1028) )             -
```

```
            INDEX   ( NAME ($INDUV..WY&USER.I)   -
                      CISZ (512) )                     -
            DATA    ( NAME ($INDUV..WY&USER.D)   -
                      FSPC (10 5) SPEED                -
            CISZ (4096) )          /*: 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 ($INDSV..WKROAD.I)   -
                      CISZ (512) )                     -
            DATA    ( NAME ($INDSV..WKROAD.D)   -
                      FSPC (10 5) SPEED                -
            CISZ (4096) )          /*: 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 (1 1)     -
                  RECSZ (1028 1028) )              -
            INDEX   ( NAME ($INDSV..WYROAD.I)   -
                      CISZ (512) )                     -
            DATA    ( NAME ($INDSV..WYROAD.D)   -
                      FSPC (10 5) SPEED                -
            CISZ (4096) )          /*: CATALOG ($VCAT) */
:/       ADD NAME=DLBVPGJ
 DELETE ($INDSV..BVPGJ) NONVSAM
:/       ADD NAME=DLBVPGR
 DELETE ($INDSV..BVPGR) NONVSAM
:/       ADD NAME=DLBVPGY
 DELETE ($INDSV..BVPGY) NONVSAM
:/       ADD NAME=DLBVPTR
 DELETE ($INDSV..BVPTR) NONVSAM
:/       ADD NAME=DLBVPQJ
 DELETE ($INDSV..BVPQJ) NONVSAM
:/       ADD NAME=VERIFTD
 VERIFY FILE (PAC7TD)
:/       ADD NAME=VERIFAN
 VERIFY FILE (PAC7AN)
:/       ADD NAME=VERIFAE
 VERIFY FILE (PAC7AE)
:/       ADD NAME=VERIFEM
 VERIFY FILE (PAC7EM)
:/       ADD NAME=VERIFGN
 VERIFY FILE (PACGGN)
:/       ADD NAME=VERIFGU
 VERIFY FILE (PACGGU)
```

```
:/        ADD NAME=VERIFSC
 VERIFY FILE (PAC7SC)
:/        ADD NAME=VERIFSG
 VERIFY FILE (PAC7SG)
:/        ADD NAME=VERIFSN
 VERIFY FILE (PAC7SN)
:/        ADD NAME=VERIFSS
 VERIFY FILE (PAC7SS)
:/        ADD NAME=VERIFSR
 VERIFY FILE (PAC7SR)
:/        ADD NAME=VERIFSP
 VERIFY FILE (PAC7SP)
:/        ADD NAME=VERIFLB
 VERIFY FILE (PAC7LB)
:/        ADD NAME=VERIFSA
 VERIFY FILE (PAC7SA)
:/     ADD NAME=REPRO
 REPRO INFILE (IN1)    OUTFILE (OU1)
:/        ADD NAME=REPROLB
 REPRO INFILE (INLB) OUTFILE (OUTLB)
:/        ADD NAME=REPROSV
 REPRO INFILE (INSV) OUTFILE (OUTSV)
:/        ADD NAME=REPROWS
 REPRO INFILE (INWS) OUTFILE (OUTWS)
:/        ADD NAME=LIBVPGJ
 LISTCAT ENTRIES ($INDSV..BVPGJ)
:/        ADD NAME=DFSVSAM8
4096,8
:/        ADD NAME=DFSVSAM9
8192,9
OPTIONS,INSERT=SEQ
:/        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=LDBVPLB

:/        ADD NAME=LDBVPSV
9999999999**999 RECORD OF LOADING DATABASE SPA
:/        ADD NAME=LDBVPWS
999999999999999999999999999999999999
:/        ADD NAME=PACCTRL
*----------------------------------------------------------------
*      *-------------------------------------------------------*
*      * THIS PDS  IS USED  BY THE  SYSTEM  TEAM. IT  CONTAINS *
*      * THE DESCRIPTION OF MACRO INSTRUCTIONS 'APPLCTN' AND   *
*      * 'TRANSACT' OF THE VA PAC SYSTEM. THEY HAVE TO BE      *
*      * DEFINED IN THE I.M.S. CONTROL REGION.                 *
*      *-------------------------------------------------------*
*
*-----------------------------------------------------------------*
* DEFINITION OF VA PAC         DATABASES            - CONTROL REGION - *
*----------------------------------------------------------------
```

```
     DATABASE ACCESS=UP,DBD=(BVPDAE,BVPDSV,BVPDLB,BVPDQJ)
     DATABASE ACCESS=UP,DBD=(BVPDWS,BVPDTR,BVPDPA,BVPDP1)
     DATABASE ACCESS=UP,DBD=(BVPDGN,BVPDGR,BVPDGU,BVPDGY,BVPDGJ)
     DATABASE ACCESS=UP,DBD=(TDAN$BASE,TDAR$BASE)
     DATABASE ACCESS=UP,DBD=(TDAJ$BASE,TDAY$BASE)
     DATABASE ACCESS=UP,DBD=(TDDC$BASE,TDD3$BASE)
     DATABASE ACCESS=UP,DBD=(BDAJ$BASE,BDPA$BASE,BDP1$BASE)
     DATABASE ACCESS=UP,DBD=(BDAN$BASE,BDAR$BASE,BDAY$BASE)
     DATABASE ACCESS=UP,DBD=(BDDC$BASE,BDD3$BASE)
*-----------------------------------------------------------------------*
* DEFINITION OF VA PAC          TRANSACTIONS        - CONTROL REGI
*-----------------------------------------------------------------
*
*    *-------------------------------------------------------------*
*    * WARNING: THE FOLLOWING PARAMETERS OF THE MACRO INSTRUCTIONS  *
*    * 'TRANSACT' CANNOT BE MODIFIED:                               *
*    * 'CODE=' , 'SEGSIZE=' , 'MODE=' , 'SPA=' , 'SNGLSEG' (IN THE  *
*    * PARAMETER 'MSGTYPE=')                                        *
*    * THE OTHER PARAMETERS HAVE TO BE ADJUSTED TO THE STANDARDS IN *
*    * USE ON THE INSTALLATION SITE.                                *
*    *-------------------------------------------------------------*
   APPLCTN  PSB=BVPRIMS
   TRANSACT CODE=$BASE,SEGSIZE=03500,MODE=SNGL,SEGNO=00050,    $Y
            PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,   $Y
            MSGTYPE=(SNGLSEG,RESPONSE,$CLS),SPA=(150)
   APPLCTN  PSB=BVPSSOC
   TRANSACT CODE=$CSOC,SEGSIZE=32000,MODE=SNGL,SEGNO=00050,    $Y
            PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
            MSGTYPE=(MULTSEG,RESPONSE,$CLS)
*-----------------------------------------------------------------------*
* DEFINITION OF VA PACBASE     BMP                  - CONTROL REGION - *
*-----------------------------------------------------------------------*
   APPLCTN  PSB=PACB$SUG,PGMTYPE=BATCH
   APPLCTN  PSB=PCMPUF$SUG,PGMTYPE=TP
:/     ADD NAME=DLPQCE
 DELETE ($INDUN..PQCE.&USER)
:/     ADD NAME=MAXKEY
99999999999999999999999999999999999999999999999999999999999999999999
:/     ADD NAME=REPRO999
 REPRO INFILE (MAXKEY)   OUTFILE (SYSPAF)
:/     ADD NAME=KEY01
00000001
:/     ADD NAME=DLWKREOR
 DELETE ($INDUV..WK&USER) CLUSTER
:/     ADD NAME=DLWYREOR
 DELETE ($INDUV..WY&USER) CLUSTER
:/     ADD NAME=DLWKROAD
 DELETE ($INDSV..WKROAD) CLUSTER
:/     ADD NAME=DLWYROAD
 DELETE ($INDSV..WYROAD) CLUSTER
:/       ADD NAME=DFBVPP1
 DELETE ($INDSV..BVPP1) CLUSTER
 DEFINE CLUSTER ( NAME ($INDSV..BVPP1)          -
                 SHR (2,3)  RUS  KEYS (37 5)       -
                 INDEXED                           -
```

```
                     VOL ($VOLSV)   CYL (1 1)            -
                     RECSZ (042 042) )                   -
          INDEX   ( NAME ($INDSV..BVPP1.I)       -
                     CISZ (1024) )                       -
          DATA    ( NAME ($INDSV..BVPP1.D)      -
                     CISZ (1024) ) /*: CATALOG ($VCAT) :*/
  :/        ADD NAME=DFBVPPA
  DELETE ($INDSV..BVPPA) CLUSTER
  DEFINE CLUSTER ( NAME ($INDSV..BVPPA)        -
                     SHR (2,3)  RUS                      -
                     NONINDEXED                          -
                     VOL ($VOLSV) CYL (5,1)              -
                     RECSZ (4089 4089) )                 -
          DATA    ( NAME ($INDSV..BVPPA.D)        -
                     FSPC (10,5)                         -
                     CISZ (4096) ) /*: CATALOG ($VCAT) :*/
  :/        ADD NAME=REPROPA
  REPRO INFILE (INPA) OUTFILE (OUTPA)
  :/        ADD NAME=LDBVPPA
  99999999999999999999999999999999999999
  :/     ADD NAME=DFBVPSA
  DELETE ($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 ($SCAT) */
  :/     ADD NAME=DFBVPGK
  DELETE ($INDSV..BVPGK) CLUSTER
    DEFINE CLUSTER ( NAME ($INDSV..BVPGK)          -
                     SHR (2) RUS      KEY (29 0)       -
                     VOL ($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) */
  PP
  //
```

Initialization

D03DEFIN module: '$prfj.D3' job

Allocation of system and administration files

Step	Program	Comments
STEP1	IDCAMS	Delete/Define GS and GN

Step	Program	Comments
DELGR	IDCAMS	Delete GR
DEFGR	IEBFR14	GR definition
DELGJ	IDCAMS	Delete GJ
DEFGJ	IEBFR14	GJ definition
DELGY	IDCAMS	Delete GY
DEFGY	IEBFR14	GY definition
DELTR	IDCAMS	Delete TR
DEFTR	IEBFR14	TR definition

Execution JCL

```
//$PRFJ.D3 JOB ($CCPT),'PAC D03DEFIN',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//********************************************************************
//* VISUALAGE PACBASE                                               *
//*                                                                 *
//* INSTALLATION                                                    *
//*    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=$INDSN..BVPSY(DFBVPGS),DISP=SHR
//       DD DSN=$INDSN..BVPSY(DFBVPGN),DISP=SHR
//       DD DSN=$INDSN..BVPSY(DFBVPGK),DISP=SHR
//*
//DELGR   EXEC PGM=IDCAMS
//SYSIN    DD DSN=$INDSN..BVPSY(DLBVPGR),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//DEFGR   EXEC PGM=IEFBR14
//PACGGR   DD DSN=$INDSV..BVPGR,UNIT=$UNITSV,
//           DISP=(,CATLG,DELETE),VOL=SER=$VOLSV,
//           DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//           SPACE=(CYL,(6,2),RLSE)
//*
//DELGJ   EXEC PGM=IDCAMS
//SYSIN    DD DSN=$INDSN..BVPSY(DLBVPGJ),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//DEFGJ   EXEC PGM=IEFBR14
//PACGGJ   DD DSN=$INDSV..BVPGJ,UNIT=$UNITSV,
//           DISP=(,CATLG,DELETE),VOL=SER=$VOLSV,
//           DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//           SPACE=(CYL,(6,2),RLSE)
//*
```

```
//DELGY   EXEC PGM=IDCAMS
//SYSIN    DD DSN=$INDSN..BVPSY(DLBVPGY),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//DEFGY   EXEC PGM=IEFBR14
//PACGGY   DD DSN=$INDSV..BVPGY,UNIT=$UNITSV,
//           DISP=(,CATLG,DELETE),VOL-SER=$VOLSV,
//           DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//           SPACE=(CYL,(6,2),RLSE)
//*
//DELTR   EXEC PGM=IDCAMS
//SYSIN    DD DSN=$INDSN..BVPSY(DLBVPTR),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//DEFTR   EXEC PGM=IEFBR14
//PACGTR   DD DSN=$INDSV..BVPTR,UNIT=$UNITSV,
//           DISP=(,CATLG,DELETE),VOL-SER=$VOLSV,
//           DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//           SPACE=(CYL,(2,1),RLSE)
//*
//
```

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 files

Step	Program	Comments
INITGS	IDCAMS	Initialization of GS extraction schema file max. key (PAF extension)

Execution JCL

```
//$PRFJ.D3I JOB ($CCPT),'PAC D03INI',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//**********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//* INSTALLATION : INITIALIZATION                                     *
//*   INITIALIZATION: JOB TO RUN ONLY                                 *
//*   THE FIRST TIME VA PAC IS INSTALLED                              *
//*    . INITIALIZATION OF PDM EXTENSION-USER FILE "GS"               *
//*    . INITIALIZATION OF GENERATION RIGHTS FILE "GK"                *
//**********************************************************************
//INITGS 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)
//*
//INITGK 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)
//*
```

Renaming the Exit users batch programs

D04MBR module: '$prfj.D4B' job

Rename Exit users batch programs

Step	Program	Comments
STEP1	IEBCOPY	RENAME load modules Exit users

Step	Program	Comments
STEP2	IEBCOPY	COPY load modules Exit users

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=((BVPTPSTB,BVPTPST))
    S M=((BVPTPWSB,BVPTPWS))
/*
//*
//STEP2    EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//I1       DD DSN=&&TEMP,DISP=(OLD,DELETE)
```

```
//01      DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//SYSIN   DD *
   C  I=((I1,R)),O=01
/*
```

Renaming the on-line PSBs

D04PSREN module: '$prfj.D4P' job

Rename on-line PSBs

Step	Program	Comments
STEP1	IEBCOPY	Rename and copy on-line PSBs

Execution JCL

```
//$PRFJ.D4P JOB ($CCPT),'PAC D04PSR',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*******************************************************************
//* VISUALAGE PACBASE                                               *
//*                                                                 *
//*            INSTALLATION - D04PSREN
//*      - RENAME PSB TP -
//*************************************************************
//STEP1    EXEC PGM=IEBCOPY,REGION=0M
//SYSPRINT DD SYSOUT=$OUT
//I1       DD DSN=$HLQ..SBVPPSB,DISP=SHR
//01       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=01
     S M=((BVPRIMST,BVPRIMS))
     S M=((BVPSSOCT,BVPSSOC))
/*
//*
//STEP2    EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//I1       DD DSN=&&TEMP,DISP=(OLD,DELETE)
//01       DD DSN=$HLQ..SBVPPSB,DISP=SHR
//SYSIN    DD *
   C  I=((I1,R)),O=01
/*
```

Loading of the PSBs and DBDs sources

D04SRCP module: loading of PSBs sources

This step must be executed if the default parameter values $BASE and $SUG
haven't been taken into account at installation.

The default values of these parameters are:

• 35 for the $SUG parameter,

- P350 for the $BASE parameter.

  In this case, the whole sources must be re-compiled.

  The '$PRFJ.SRCP' JOB allocates the $INDUN..$BASE.SRCP library and updates, via IEBUPDTE, the PSBs sources in this library.

  The PSBs sources are cataloged under the following name:
- xxxxxxyy with the $SUG parameter value for yy and the VA Pac program name for xxxxxx. These are the Batch PSBs.
- BVPyyyy with the RIMS value for yyyy.

  This is the on-line PSB.
- BVPyyyy with the SSOC value for yyyy.

  This is the PSB, which is used for the Socket transaction.

D04SRCD module: loading of DBDs sources

This step must be executed if the default parameter values $BASE and $SUG haven't been taken into account at installation.

The $BASE default parameter value is: P350.

In this case, the whole sources must be re-compiled.

The '$PRFJ.SRCD' JOB allocates the $INDUN..$BASE.SRCD library and updates via IEBUPDTE the DBDs sources in this library.

The DBDs sources are cataloged under the name:
- BDxxyyyy
- TDxxyyyy

  with xx equal to the database code and yyyy equal to the chosen $BASE parameter value.

Compilation of ACB

D04ACB module: ACB compilation

In this step, you find the job '$PRFJ.ACB' which is made to build all the ACB necessary for the use of VA Pac system in on-line mode.

Execution JCL

```
//$PRFJ.ACB JOB ($CCPT),'ACBGEN',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($PRCLIB)
//ET010   EXEC ACBGEN,SOUT='$OUT'
//********************************************************
//*               COMPILATION OF ACB                    *
//*               -----------------                      *
```

```
//*        THIS JOB MUST BE EXECUTED AFTER LOADING DBDLIB    *
//*        AND PSBLIB (EITHER BY LOADING OBJECT MODULES       *
//*        OR BY COMPILING DBD AND PSB).                      *
//*************************************************************
//G.SYSIN DD *
 BUILD DBD=(BVPDGR,BVPDGN,BVPDGU,BVPDGJ,BVPDGY)
 BUILD DBD=(BVPDAE,BVPDSV,BVPDLB,BVPDWS,BVPDTR)
 BUILD DBD=(BVPDPA,BVPDP1,BVPDQJ)
 BUILD DBD=(TDAN$BASE,TDAR$BASE,TDAJ$BASE,TDAY$BASE)
 BUILD DBD=(BDAN$BASE,BDAR$BASE,BDAJ$BASE,BDAY$BASE)
 BUILD DBD=(TDDC$BASE,TDD3$BASE)
 BUILD DBD=(BDPA$BASE,BDP1$BASE)
 BUILD PSB=(BVPRIMS,BVPSSOC)
 BUILD PSB=(PACB$SUG,PCMPUF$SUG)
//G.IMSACB DD DSN=$ACBLIB,DISP=SHR
//G.IMS    DD DSN=$PSBLIB,DISP=SHR
//         DD DSN=$DBDLIB,DISP=SHR
```

Loading of Procedures

It is recommended that all operation procedures be cataloged in one procedure 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 procedure 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 : ALLOCATION                                         *
//*                                                                 *
//*******************************************************************
//*
//STEP1  EXEC PGM=IEFBR14
```

```
//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 the 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 older 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=BVPCHPM
:/ 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=BVPGPPM
:/ ADD NAME=BVPGPRP
:/ ADD NAME=BVPGPRC
:/ ADD NAME=BVPGPRT
:/ ADD NAME=BVPGPRU
:/ 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=BVPLDDC
:/ ADD NAME=BVPLDLB
:/ ADD NAME=BVPLDPA
:/ ADD NAME=BVPLDP1
:/ ADD NAME=BVPLDSV
:/ ADD NAME=BVPLDWS
:/ ADD NAME=BVPMB25
:/ ADD NAME=BVPMB30
:/ 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=BVPPPAD
:/ ADD NAME=BVPPP25
:/ ADD NAME=BVPPQCA
:/ ADD NAME=BVPPQCE
:/ ADD NAME=BVPPRGS
:/ ADD NAME=BVPREOR
:/ ADD NAME=BVPREST
:/ ADD NAME=BVPRESY
:/ ADD NAME=BVPRMTD
:/ ADD NAME=BVPROAD
```

```
:/ 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=BVPUPDP
:/ ADD NAME=BVPUPDT
:/ ADD NAME=BVPUPGP
:/ 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=$HLQ..SBVPF5(BVPSA$LSK),DISP=SHR
//SCI      DD DSN=$HLQ..SBVPF6(BVPSC$LSK),DISP=SHR
//SGI      DD DSN=$HLQ..SBVPF5(BVPSG$LSK),DISP=SHR
//SNI      DD DSN=$HLQ..SBVPF5(BVPSN$LSK),DISP=SHR
//SRI      DD DSN=$HLQ..SBVPF5(BVPSR$LSK),DISP=SHR
//SSI      DD DSN=$HLQ..SBVPF5(BVPSS$LSK),DISP=SHR
//SPI      DD DSN=$HLQ..SBVPF5(BVPSP),DISP=SHR
//SYSIN    DD *
  REPRO INFILE (SAI)   OUTFILE (SAO)
  REPRO INFILE (SCI)   OUTFILE (SCO)
  REPRO INFILE (SGI)   OUTFILE (SGO)
  REPRO INFILE (SNI)   OUTFILE (SNO)
  REPRO INFILE (SRI)   OUTFILE (SRO)
  REPRO INFILE (SSI)   OUTFILE (SSO)
  REPRO INFILE (SPI)   OUTFILE (SPO)
//*
//STEP3  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD *
 DELETE ($INDSN..BVPSF)
//*
//STEP4  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=$OUT
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=$HLQ..SBVPF7(BVPSF),DISP=SHR
//SYSUT2   DD DSN=$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 on-line help

D07AE0 module: '$prfj.D7' job

This JCL defines, and loads the AE error messages files.

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

The $ line is required. It enable to recognize the language of generation
skeletons chosen at installation time ($LSK parameter). This line should not be
modified.

The second line is meant to choose the on-line help and error messages
language.

English is the default language code.

To recognize one or more other languages, you must add as many parameter
lines containing the language code on two characters in position 3.

To recognize all the available languages, you must enter a parameter line
containing '**' in position 3.

Code	Language name
DE	German
EN	English
ES	Spanish
FR	French
PT	Portuguese
**	All languages

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

Execution JCL

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

Initialization of specific files

D07INIT module: '$prfj.I7' job

This JCL defines and then initializes the following databases:

- The SV spas database (used to save dialogs),
- The LB jobs follow up database,
- The PAF TP PA working database,
- The PUF WS working database.

  It must be executed each time a version is re-installed and each time the re-installation of these databases is necessary. Each database can yet be independently re-initialized by executing the following procedures:

- LDLB for the LB database
- LDSV for the SV database
- LDWS for the WS database
- LDP1 for the PA database !

Step	Program	Comments
STEP1	LDLB	Initialization of LB
STEP2	LDSV	Initialization of SV
STEP3	LDP1	Initialization of PA
STEP4	LDWS	Initialization of WS

Execution JCL

```
//$PRFJ.I7 JOB ($CCPT),'AE BASE DEF',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//*********************************************************************
//* VISUALAGE PACBASE                                                 *
//*                                                                   *
//* INSTALLATION : INITIALIZATION OF SPECIFICS FILES
//*************************************************************
//STEP1  EXEC BVPLDLB
```

```
//*
//STEP2  EXEC BVPLDSV
//*
//STEP3  EXEC BVPLDP1
//*
//STEP4  EXEC BVPLDWS
//*
```

Installation of the Administration Database

Users File Initialization

D08INGU module: '$prfj.D8' job

This JCL should be submitted at 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 when the test JCL lines are run.

Note:

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

Step	Program	Comments
INGU	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 at the first installation only.

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

Caution

If the files are handled on SMS, delete the lines DD //GDGMOD of the IDCAMS Steps JCL before submitting the job.

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

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..SBVPF2(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 at 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. With the exit of this stage, a key of evaluation is installed, but allows only the execution of certain procedures, in particular the procedures of installation.

The access key must be updated via the Administrator workbench, to take into account the whole product.

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 must be submitted at the first installation only.

It defines then loads the QJ 'archival journal file of VisualAge Pacbase interface and configuration management'.

Step	Program	Comments
OSAMQJ1	IDCAMS	DELETE de QJ
OSAMQJ2	IEBFR14	DEFINE de QJ
PCMINJ	PCMINJ	QJ initialization

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 at 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..SBVPF8(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

This step is recommended to avoid, in case of problem, entering again the access key.

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

Security Systems Interface

Introduction

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

The Interface is designed to ensure control communication between the security systems and VA Pac.

In order to ensure compatibility with all security systems, the VA Pac system is not directly connected 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 implementation of controls between the security system and VA Pac, two types of operations are required, the first ones must be done in the Administrator workbench module, and the others in the security system.

- From the Administrator workbench module

  You have to 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 realize 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 on 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.

Implementation for RACF

Installing and using RACF implies that it is possible to 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 a resource class

  The definition of a resource class is made 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 resource authorizations

  It must be done with PERMIT used as follows:

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

  PERMIT nBBB CLASS(cccc) ID(user) ACCESS(ALTER) n varies from 0 to 3 and corresponds to the VA Pac authorization level for the BBB library.

ON-LINE MODE REQUESTS

In on-line mode, a command allows you to execute a RACF request and to declare the user access authorizations to the resources ('CALL CBLTDLI USING AUTH' in the BVPRACF control sub-program).
- Connection in on-line mode

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

  They are retrieved in the PCB I/O while connecting to the IMS transaction.

  All authorization controls are made for this user via the RACF request.
- Workbench connection

  On the sign-on screen, only the user and password of the middleware are required.

As for the on-line mode, the user code is retrieved in the PCB I/O.

– Use of the CPIC protocol: the user validity is checked (equivalent to CESN).

– Use of sockets (SPECIFIC TO CICS): it implies the use of a CICS listener, which in standard, is deprived from authentication system. The RACF control is performed via the security exit-user BVPCICSE delivered by SMP/E in the PDS hlq.SBVPMTR8.

– Implementation of the security exit-user BVPCICSE.

It must be declared with the clause 'EXECKEY(CICS)' in the CSD (see the installation JCL D00CSD).

On the CICS, enter the 'EZAC AL' command:

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

After selecting the listener:

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

You find the following screen where you must enter the exit-user code:

```
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 GIVESOCKET
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    ===>            ===>
```

BATCH REQUESTS

Unlike the on-line mode, there is no command allowing the execution of a RACF request: therefore, it is necessary to execute the RACF request through the use of an assembler sub-program.

This sub-program BVPSECRA, supplied by SMP/E in the PDS hlq.SBVPMBR8, must be copied in an authorized program library (AUTHORIZED LIBRARY) with the name BVPSECUR and a RACLIST authorization. It must be declared in the ICHAUTAB table with authorization of RACLIST SVC and RACINIT SVC (X'C0000000').

Then, you must check if the user is authorized to execute the procedure. In case of a multi-user procedure (UPDT for instance), for each identification line met, only authorizations to library levels are checked.

Implementation for TOPSECRET

The sources of the sub-programs (assembler) for 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 its compilation and link-edit.

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 made up of 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 used to declare an administrator.

DEFINITION OF RESOURCES
- Definition of a resource class
  SS ADD (RDT) RESCLASS(cccc) RESCODE(xx)
  cccc = code of the resource class corresponding 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)

Installation of the Development Database

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

You do so via the 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 initialized.

Allocation and Loading of Database Parameters

I01SY module: '$prfj.I1' job

NOTE: This PDS contains the definition of the Database files. A default size is specified for the test database supplied at installation. As the 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: Change :/ into ./ 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
            //*********************************************************************
            //*             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 (3)    RUS   KEYS (49,6)    -
                             VOL ($VOLUV)                    -
                             CYL (20 1)                      -
                             RECSZ (68,68) )                 -
                    INDEX   ( NAME ($INDUV..$BASE.AN.I)      -
                             CISZ (4096) )                   -
                    DATA    ( NAME ($INDUV..$BASE.AN.D)      -
                             FSPC (10,5)                     -
                    CISZ (4096) )         /*: CATALOG ($VCAT) */
            :/     ADD NAME=DL$BASE.MY
             DELETE ($INDUV..$BASE.MY) NONVSAM
            :/     ADD NAME=LI$BASE.AJ
             LISTCAT ENTRIES ($INDUV..$BASE.AJ)
            :/       ADD NAME=DF$BASE.D3
             DELETE ($DSMS..$BASE.D3) CLUSTER
             DEFINE CLUSTER ( NAME ($DSMS..$BASE.D3)          -
                             SHR (2,3)  RUS  KEYS (31 5)       -
                             INDEXED                           -
                             VOL ($VOLUV)      TRK (10 5)       -
                             RECSZ (036 036) )                 -
                    INDEX   ( NAME ($DSMS..$BASE.D3.I)         -
                             CISZ (1024) )                     -
                    DATA    ( NAME ($DSMS..$BASE.D3.D)         -
                             CISZ (1024) ) /*: CATALOG ($VCAT) :*/
            :/       ADD NAME=DF$BASE.DC
             DELETE ($DSMS..$BASE.DC) CLUSTER
             DEFINE CLUSTER ( NAME ($DSMS..$BASE.DC)          -
                             SHR (2,3)  RUS                    -
```

```
                      NONINDEXED                          -
                      VOL ($VOLUV)      CYL (2 1)         -
                      RECSZ (4089 4089) )                 -
          DATA    ( NAME ($DSMS..$BASE.DC.D)      -
                      FSPC (10,5)                        -
                      CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/        ADD NAME=DFTABTDF
 DELETE ($TABTDF) CLUSTER
 DEFINE CLUSTER ( NAME ($TABTDF)                          -
                      SHR  (2,3)   KEYS (21,0)            -
                      VOL ($VOLUV)  CYL (1,1)             -
                      RECSZ (240,240) RUS )               -
          INDEX   ( NAME ($TABTDF..I)                 -
                      CISZ (1024) )                      -
          DATA    ( NAME ($TABTDF..D)                 -
                      FSPC (10,5)                        -
                      CISZ (2048) ) /*: CATALOG ($VCAT) :*/
:/        ADD NAME=DF$BASE.P1
 DELETE ($INDUV..$BASE.P1) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.P1)       -
                      SHR (2,3)  RUS  KEYS (37 5)        -
                      INDEXED                            -
                      VOL ($VOLUV)     CYL (1 1)         -
                      RECSZ (042 042) )                  -
          INDEX   ( NAME ($INDUV..$BASE.P1.I)     -
                      CISZ (1024) )                      -
          DATA    ( NAME ($INDUV..$BASE.P1.D)     -
                      CISZ (1024) ) /*: CATALOG ($VCAT) :*/
:/        ADD NAME=DF$BASE.PA
 DELETE ($INDUV..$BASE.PA) CLUSTER
 DEFINE CLUSTER ( NAME ($INDUV..$BASE.PA)       -
                      SHR (2,3)  RUS                     -
                      NONINDEXED                         -
                      VOL ($VOLUV) CYL (5,1)             -
                      RECSZ (4089 4089) )                -
          DATA    ( NAME ($INDUV..$BASE.PA.D)     -
                      FSPC (10,5)                        -
                      CISZ (4096) ) /*: CATALOG ($VCAT) :*/
:/        ADD NAME=DL$BASE.AJ
 DELETE ($INDUV..$BASE.AJ) NONVSAM
:/        ADD NAME=DL$BASE.AY
 DELETE ($INDUV..$BASE.AY) NONVSAM
:/        ADD NAME=DL$BASE.AR
 DELETE ($INDUV..$BASE.AR) NONVSAM
:/        ADD NAME=BL$BASE.PC
   DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUV..$BASE.PC) LIMIT (3) SCR)
:/        ADD NAME=BL$BASE.PD
   DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUV..$BASE.PD) LIMIT (3) SCR)
:/        ADD NAME=BL$BASE.PY
   DEFINE GENERATIONDATAGROUP -
         (NAME ($INDUV..$BASE.PY) LIMIT (3) SCR)
:/        ADD NAME=BL$BASE.PJ
   DEFINE GENERATIONDATAGROUP -
```

```
                (NAME ($INDUV..$BASE.PJ) LIMIT (3) SCR)
:/          ADD NAME=LD$BASE.DC
9999999999999999999999999999999 RECORD LOADING DATABASE DC (CGI)
:/          ADD NAME=LD$BASE.PA
99999999999999999999999999999999999 RECORD OF LOADING 'PA'
:/       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=REPRODC
 REPRO INFILE (INDC) OUTFILE (OUTDC)
:/        ADD NAME=REPROPA
 REPRO INFILE (INPA) OUTFILE (OUTPA)
PP
//
```

Initialization of Generation Data Groups

These jobs must be executed every time a new database is initialized.

I03DEF module: '$prfj.I3' job

Definition of the database files

Step	Program	Comments
STEP1	IDCAMS	Delete/Define AN
DELAR	IDCAMS	Delete AR
DEFAR	IEBFR14	Define AR

Step	Program	Comments
DELAJ	IDCAMS	Delete AJ
DEFAJ	IEBFR14	Define AJ
DELAY	IDCAMS	Delete AY
DEFAY	IEBFR14	Define AY
INITDC	LDDC	Initializes DC
INITPA	LDPA	Initializes PA

Execution JCL

```
//$PRFJ.I3  JOB ($CCPT),'PAC DEF',CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
// JCLLIB ORDER=($BIBP)
//**********************************************************************
//* VISUALAGE PACBASE                                                  *
//*                                                                    *
//*             INSTALLATION - DEFD                                    *
//*    DEFINITION OF THE VSAM FILES FOR ONE VA PAC DATABASE            *
//*     STEP1          : DELETE/DEFINE OF AN FILE                      *
//*     DELAR / DEFAR  : DELETE/DEFINE OF AR FILE                      *
//*     DELAJ / DEFAJ  : DELETE/DEFINE OF AJ FILE                      *
//*     DELAY / DEFAY  : DELETE/DEFINE OF AY FILE                      *
//*     INITDC         : DEFINITION AND INITILIZATION OF DC FILE       *
//*     INITPA         : DEFINITION AND INITILIZATION OF PA FILE       *
//**********************************************************************
//*
//STEP1  EXEC PGM=IDCAMS
//*:STEPCAT  DD  DSN=$VCAT,DISP=SHR
//SYSPRINT DD  SYSOUT=$OUT
//SYSIN DD DSN=$INDUN..$BASE.SY(DF$BASE.AN),DISP=SHR
//*
//DELAR   EXEC PGM=IDCAMS
//SYSIN    DD DSN=$INDUN..$BASE.SY(DL$BASE.AR),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//DEFAR   EXEC PGM=IEFBR14
//PAC7AR   DD DSN=$INDUV..$BASE.AR,UNIT=$UNITUV,
//          DISP=(,CATLG,DELETE),VOL=SER=$VOLUV,
//          DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//          SPACE=(CYL,(6,2),RLSE)
//*
//DELAJ   EXEC PGM=IDCAMS
//SYSIN    DD DSN=$INDUN..$BASE.SY(DL$BASE.AJ),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//DEFAJ   EXEC PGM=IEFBR14
//PAC7AJ   DD DSN=$INDUV..$BASE.AJ,UNIT=$UNITUV,
//          DISP=(,CATLG,DELETE),VOL=SER=$VOLUV,
//          DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//          SPACE=(CYL,(6,2),RLSE)
```

```
//*
//DELAY    EXEC PGM=IDCAMS
//SYSIN    DD DSN=$INDUN..$BASE.SY(DL$BASE.AY),DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//*
//DEFAY    EXEC PGM=IEFBR14
//PAC7AY   DD DSN=$INDUV..$BASE.AY,UNIT=$UNITUV,
//            DISP=(,CATLG,DELETE),VOL=SER=$VOLUV,
//            DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096),
//            SPACE=(CYL,(6,2),RLSE)
//*
//INITDC   EXEC BVPLDDC
//*
//INITPA   EXEC BVPLDPA
//*
```

I03INI module: '$prfj.I3I' job

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

Caution

If the files are handled under 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..SBVPF2(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 with 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 and do not add any change.

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 at the first installation and at each re-installation
of Development Database.

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..SBVPF8(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 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 from scratch FO file
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
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)
/*
//
```

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 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).
- 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 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 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.

   > **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.htm

Under 'Self Help' and 'Download', click the 'VisualAge Pacbase downloads' link, then 'VisualAge Pacbase Downloads in English', you will find a number of specific instructions, follow the 'Open Jade' and 'Tidy' links, accordingly.

Open Jade must be installed in the 'adworkbench' directory and Tidy in the 'adworkbench\workstation\publish' directory.

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.

NON-STANDARD LABELS FILE (GPMON)

To enable the WorkStation to take into account non- standard files' labels in output of a personalized generation and print procedure, you need to create a file GPMON.LBL in which you enter the specific labels of the files.

The file is to be created under the BASE directory and must contain the label and code of each specific file. Their extensions must be consistent with the usual standard extensions: Ix, Gx, Ox ou Dx (example, IP, GA...).

Each label begins in column 4. The lines are maximum 60 characters long.

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.

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

MQ-CICS Bridge.

Version of the MQCIH structure: 1 (default) or 2, depending on the used CICS Bridge interface. (1 character maximum).

- IXO_LOCALCCSID

MQ-CICS Bridge and MQ-IMS Bridge.

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.

(9 characters maximum).

- IXO_QUEUEMANAGER

  MQ-CICS Bridge and MQ-IMS Bridge.

  Local Queue Manager name of the client part.

  (4 characters maximum).

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

  Name of RACF group for IMS Connect.
- IXO_REPLYQUEUE

  MQ-CICS Bridge and MQ-IMS Bridge.

  Name of reply messages Queue.

  (48 characters maximum).
- IXO_REQUESTEXPIRY

  MQ-CICS Bridge and MQ-IMS Bridge.

  Expiration delay of the request message, in seconds.

  (9 characters maximum).
- IXO_REQUESTQUEUE

  MQ-CICS Bridge and MQ-IMS Bridge.

  Name of request messages Queue.

  (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 parameters in 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' an already installed component.
- 'Modify' i.e. add a new sub-component to an already installed component.

  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' an already installed component.

  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 on-line 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 to the version installed.

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

Your context may be :

Case 1: Standard re-installation

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

1. D04MBR: $prfj.D4B job: renaming exit-users batch load-modules,
2. D04PSREN: $prfj.D4P job: renaming on-line PSB,
3. D04SRCD: $prfj.D4SR job: DBD sources update + recompilation of DBD if necessary,
4. D04SRCP: $prfj.D4SR job: PSB sources update + recompilation of PSB if necessary,
5. D06SKEL: loading skeleton files,
6. D07AE0 : loading error messages,
7. D08XMET: installation of the Administration Model, (see the description in the appendix at the end of the manual).
8. 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).

If the D08XMET JCL report 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 I05META JCL report 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 note provided in the supply.

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.

Screen branching:

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:
    - access key entering,
    - 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 through 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 (potential) 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.

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 the 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
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7MB	&&PE25MB	Input	User input
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&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
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
BVP7AE	&INDSV..BVPAE	Input	Error labels
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Transactions formatting (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
BVP8GR	&INDSV..BVPGR	Output	Administration Database Data file
BVP8GN	&INDSV..BVPGN	Output	Administration Database Index file
BVP8GY	&INDSV..BVPGY	Output	Administration Database extension
BVP8GJ	&INDSV..BVPGJ	Output	Administration Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS file of Development Database elements

Code	Physical name	Type	Label
B7D3$BASE	DUMMY	Input	DSMS index 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='$SCAT',         VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',    LIBRARY OF LOAD-MODULES
//         OUTL=$OUT,                   OUTPUT CLASS OF REPORTS
//         OLDPE=,                         DS NAME OF OLD PE
//         UWK=$UWK,                               WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)',      TRANSACTION SPACE
//         PSBLIB='$PSBLIB',            LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//         RESLIB='$RESLIB',            IMS RESLIB
//         PROCLIB='$PRCLIB',           IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
```

```
//****************************************************************
//INPUT  EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PE25MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*--------------------------------------------------------------
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*--------------------------------------------------------------
//PTU920  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU920,PAAFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   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)
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7ET   DD SYSOUT=&OUT
```

```
//*--------------------------------------------------------------
//PAF900  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPAF900,PAAFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=(0,NE,PTU920)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//*--------------------------------------------------------------
//PACA15  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPACA15,PAAFIM$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
```

```
//          COND=((0,NE,PTU920),(0,NE,PAF900))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7DC$BASE  DD DUMMY
//B7D3$BASE  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
//*
```

Retrieval of the Development Database (PC25)

PC25 - Introduction

<u>Principle</u>

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

<u>Execution conditions</u>

None.

<u>Printed output</u>

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.

<u>Result</u>

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

<u>Splitting up of the comment description (-G)</u>

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'.

Long data: user entities, input aids, report layouts

There are no more continuation records for these entities. There is only one index for one main record and one index for each continuation record. Long data are 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 at 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)
BVP7AE	&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)
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
BVP7AE	&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

Code	Physical name	Type	Label
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

Code	Physical name	Type	Label
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
BVP7AE	&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
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
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7PZ	&&PZ	Input	User Entities (2.5 release)
PAC7PB	&&PC	Input	First data record

Code	Physical name	Type	Label
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
BVP7AE	&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.)
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 long data

Sort criteria: SRTPC25 member of the SY PDS

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

Code	Physical name	Type	Label
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
BVP7AE	&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
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=$BASE,              CODE OF VAPAC DATABASE
//         INDSN='$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',           VA PAC 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)',  VA PAC DATABASE BACKUP 2
```

```
//          CYL='(10,1)',                          TEMPORARY SPACE
//          PSBLIB='$PSBLIB',            LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//          RESLIB='$RESLIB',            IMS RESLIB
//          PROCLIB='$PRCLIB',           IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//INPUT   EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PC25MB,DISP=(,PASS),
//            UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*--------------------------------------------------------------
//PTU911  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTU911,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB   DD DSN=&&PC25MB,DISP=(OLD,DELETE)
//BVP7AE   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
//*--------------------------------------------------------------
//PTU909  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU909,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=(0,NE,PTU911)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//              DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//              BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//              BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   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)
//*--------------------------------------------------------------
//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,PASS)
//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=$BCOB,DISP=SHR
//SYSOUT    DD SYSOUT=&OUT
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//SYSOUX    DD SYSOUT=&OUT
//PAC7PB    DD DSN=&&PC,DISP=(OLD,PASS)
//PAC7PK    DD DSN=&&PK,DISP=(OLD,DELETE)
//PAC7PD    DD DSN=&&PD,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(TRK,1,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7KP    DD DSN=&&KP,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//SORTWK01  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03  DD UNIT=&UWK,SPACE=(CYL,&CYL,,CONTIG)
//SYSUDUMP  DD SYSOUT=&OUT
//*-----------------------------------------
//PTU919  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU919,PACFIC$SUG,&BUF,
```

```
//            &SPIE&TEST&EXCPVR&RST,&PRLD,
//            &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//            &IRLM),
//            COND=(0,NE,PTU911)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//            DD DSN=&STEPLIB,DISP=SHR
//            DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//            DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7PB   DD DSN=&&PD,DISP=(OLD,DELETE)
//PAC7PE   DD DSN=&&PE,DISP=(OLD,DELETE)
//PAC7PH   DD DSN=&&PH,DISP=(OLD,DELETE)
//PAC7PM   DD DSN=&&PM,DISP=(OLD,PASS)
//PAC7KP   DD DSN=&&KP,DISP=(OLD,DELETE)
//PAC7PD   DD DSN=&&PB,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PO   DD DSN=&&PO,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7ZP   DD DSN=&&EP,DISP=(NEW,PASS),UNIT=&UWK,
//            SPACE=(CYL,&CYL,RLSE),
//            DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//*------------------------------------------------------------
//PTU912   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTU912,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=(0,NE,PTU911)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//            DD DSN=&STEPLIB,DISP=SHR
//            DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
```

```
//           DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT
//DDSNAP    DD SYSOUT=&OUT
//PROCLIB   DD DSN=&PROCLIB,DISP=SHR
//IEFRDER   DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON    DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE    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)
//*-------------------------------------------------------------
//PTU913  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU913,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=(0,NE,PTU911)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT    DD SYSOUT=&OUT
//SYSOUX    DD SYSOUT=&OUT

//DDSNAP    DD SYSOUT=&OUT
//PROCLIB   DD DSN=&PROCLIB,DISP=SHR
//IEFRDER   DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON    DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB   DD DSN=&SORTLIB,DISP=SHR
```

```
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   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)
//*-------------------------------------------------------------
//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=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU914,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=(0,NE,PTU911)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   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
//*
```

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
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
BVP7AE	&INDSV..BVPAE	Input	Error labels
PAC7MB	&&PG20MB	Input	User Entities
PAC7GY	&&PACGY	Output	Generation-print commands transactions (length=310)

Code	Physical name	Type	Label
PAC7GZ	&&PACGZ	Output	JCL lines transactions (length = 310)
PAC7ET		Report	Error report

Transactions formatting: PAF900

Code	Physical name	Type	Label
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN$BASE	&INDUV..&BASE.AN	Input	Development Database index
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&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
B7AR$BASE	&INDUV..&BASE.AR	Output	Development Database Data file
B7AN$BASE	&INDUV..&BASE.AN	Output	Development Database index
B7AY$BASE	&INDUV..&BASE.AY	Output	Development Database extension
B7AJ$BASE	&INDUV..&BASE.AJ	Output	Development Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file

Code	Physical name	Type	Label
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS file of Development Database Elements
B7D3$BASE	DUMMY	Input	Index 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
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
BVP7AE	&INDSV..BVPAE	Input	Error labels
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions

Code	Physical name	Type	Label
PAC7MV	&&PAC7MV	Output	Transactions formatting (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
BVP8GR	&INDSV..BVPGR	Output	Administration Database Data file
BVP8GN	&INDSV..BVPGN	Output	Administration Database Index file
BVP8GY	&INDSV..BVPGY	Output	Administration Database extension
BVP8GJ	&INDSV..BVPGJ	Output	Administration Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS file of Development Database elements
B7D3$BASE	DUMMY	Input	DSMS index 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	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=$BASE,               CODE OF VAPAC DATABASE
//         OUT=$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='$SCAT',            VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',    LIBRARY OF LOAD-MODULES
//         OUTL=$OUT,                   OUTPUT CLASS OF REPORTS
//         OLDPG=,                         DS NAME OF OLD PG
//         UWK=$UWK,                              WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)',      TRANSACTION SPACE
//         PSBLIB='$PSBLIB',            LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//         RESLIB='$RESLIB',            IMS RESLIB
//         PROCLIB='$PRCLIB',           IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//************************************************************
//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
//PAC7AN   DD DSN=&INDUV..&BASE.AN,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(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*----------------------------------------------------------------
//PTU908 EXEC PGM=BVPTU908
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//PAC7IN   DD DSN=&OLDPG,DISP=SHR
//PAC7OU   DD DSN=&&PG,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(10,5),RLSE),
//            DCB=(RECFM=FB,LRECL=150,BLKSIZE=6150)
//*----------------------------------------------------------------
//PTU921  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU921,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   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=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPAF900,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=(0,NE,PTU921)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//*-------------------------------------------------------------
//PACA1Y  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPACA15,PACFIM$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
```

```
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=((0,NE,PTU921),(0,NE,PAF90Y))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//         DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE  DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE  DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7DC$BASE  DD DUMMY
//B7D3$BASE  DD DUMMY
//PAC7IE   DD SYSOUT=&OUTL
//PAC7IF   DD SYSOUT=&OUTL
//PAC7ME   DD DSN=&&PAC7ME,DISP=(OLD,DELETE)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(OLD,DELETE)
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//*-------------------------------------------------------------
//PAF90Z   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPAF900,PAAFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=(0,NE,PTU921)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
```

```
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//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=27676)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//*-------------------------------------------------------------
//PACA1Z   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPACA15,PAAFIM$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=((0,NE,PTU921),(0,NE,PAF90Z))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
```

```
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//             BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//             BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7DC$BASE DD DUMMY
//B7D3$BASE 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
//*
```

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
BVP7AE	&INDSV..BVPAE	Input	Error labels
PAC7MB	&&PG25MB	Input	User Entities
PAC7GY	&&PACGY	Output	Generation-print commands transactions (length=310)

Code	Physical name	Type	Label
PAC7GZ	&&PACGZ	Output	JCL lines transactions (length = 310)
PAC7ET		Report	Error report

Transactions formatting: PAF900

Code	Physical name	Type	Label
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN$BASE	&INDUV..&BASE.AN	Input	Development Database index
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&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
B7AR$BASE	&INDUV..&BASE.AR	Output	Development Database Data file
B7AN$BASE	&INDUV..&BASE.AN	Output	Development Database index
B7AY$BASE	&INDUV..&BASE.AY	Output	Development Database extension
B7AJ$BASE	&INDUV..&BASE.AJ	Output	Development Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file

Code	Physical name	Type	Label
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS file of Development Database Elements
B7D3$BASE	DUMMY	Input	Index 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
BVP8GR	&INDSV..BVPGR	Input	Administration Database data
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
BVP7AE	&INDSV..BVPAE	Input	Error labels
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions

Code	Physical name	Type	Label
PAC7MV	&&PAC7MV	Output	Transactions formatting (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
BVP8GR	&INDSV..BVPGR	Output	Administration Database Data file
BVP8GN	&INDSV..BVPGN	Output	Administration Database Index file
BVP8GY	&INDSV..BVPGY	Output	Administration Database extension
BVP8GJ	&INDSV..BVPGJ	Output	Administration Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS file of Development Database elements
B7D3$BASE	DUMMY	Input	DSMS index 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	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=$BASE,              CODE OF VAPAC DATABASE
//         OUT=$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='$SCAT',          VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',    LIBRARY OF LOAD-MODULES
//         OUTL=$OUT,                OUTPUT CLASS OF REPORTS
//         OLDPG=,                        DS NAME OF OLD PG
//         UWK=$UWK,                             WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)',      TRANSACTION SPACE
//         PSBLIB='$PSBLIB',            LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//         RESLIB='$RESLIB',            IMS RESLIB
//         PROCLIB='$PRCLIB',           IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//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
//PAC7AN    DD DSN=&INDUV..&BASE.AN,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(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*----------------------------------------------------------------
//PTU921   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTU921,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//         DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MB   DD DSN=&&PG25MB,DISP=(OLD,DELETE)
//PAC7PG   DD DSN=&OLDPG,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//         SPACE=(TRK,(100,20),RLSE),
//         DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7GZ   DD DSN=&&PACGZ,DISP=(,PASS),UNIT=&UWK,
//         SPACE=(TRK,(100,20),RLSE),
//         DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//*----------------------------------------------------------------
//PAF90Y   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPAF900,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=(0,NE,PTU921)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
```

```
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//           SPACE=&SPAMB,
//           DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//*-------------------------------------------------------------
//PACA1Y  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPACA15,PACFIM$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=((0,NE,PTU921),(0,NE,PAF90Y))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
```

```
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7DC$BASE   DD DUMMY
//B7D3$BASE   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
//*-------------------------------------------------------------
//PAF90Z  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPAF900,PAAFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=(0,NE,PTU921)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
```

```
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//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=27676)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//            SPACE=&SPAMB,
//            DCB=(RECFM=FB,LRECL=748,BLKSIZE=27676)
//*------------------------------------------------------------
//PACA1Z   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPACA15,PAAFIM$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=((0,NE,PTU921),(0,NE,PAF90Z))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GY   DD DSN=&INDSV..BVPGY,DISP=SHR
```

```
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7DC$BASE DD DUMMY
//B7D3$BASE 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
//*
```

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 archival file in the new version
format.

Execution conditions

None.

Printed output

This procedure generates a transaction file indicating the number of
transactions retrieved in their initial format 2.n, 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 conversion from a 2.0 or 2.5 version into the new version.

Sometimes, to retrieve some transactions, it is necessary to search for information in the new version Database. But it is possible that such information 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
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AY$BASE	&INDUV..&BASE.AY	Input	Development Database extension data
PAC7JP	&&NEWPJ	Output	Journal file in the new format (length=170)
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
//         OLDPJ=,                         DS NAME OF OLD PJ
//         UWK=$UWK,                             WORK UNIT
//         PSBLIB='$PSBLIB',          LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',          LIBRARY OF DBD'S
//         RESLIB='$RESLIB',          IMS RESLIB
//         PROCLIB='$PRCLIB',         IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
```

```
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//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
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//*--------------------------------------------------------------
//PTU918  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU918,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,&IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//         DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//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

<u>Principle</u>

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 produces a report indicating 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

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)
BVP7AE	&INDSV..BVPAE	Input	Error message
PAC7MB	&&PP25MB	Input	User input
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN$BASE	&INDUV..&BASE.AN	Input	Development Database index
B7AY$BASE	&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
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN$BASE	&INDUV..&BASE.AN	Input	Development Database index
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&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
B7AR$BASE	&INDUV..&BASE.AR	Output	Development Database Data file
B7AN$BASE	&INDUV..&BASE.AN	Output	Development Database index
B7AY$BASE	&INDUV..&BASE.AY	Output	Development Database extension
B7AJ$BASE	&INDUV..&BASE.AJ	Output	Development Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS file of Development Database Elements
B7D3$BASE	DUMMY	Input	Index 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

PP25 - Execution JCL

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

```
//*              RETRIEVAL OF PP FILE
//*
//* -------------------------------------------------------
//*
//BVPPP25  PROC BASE=$BASE,                CODE OF VAPAC DATABASE
//         OUT=$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='$SCAT',             VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//         OUTL=$OUT,                    OUTPUT CLASS OF REPORTS
//         OLDPP=,                             DS NAME OF OLD PP
//         UWK=$UWK,                                     WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)',      TRANSACTION SPACE
//         PSBLIB='$PSBLIB',             LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',             LIBRARY OF DBD'S
//         RESLIB='$RESLIB',             IMS RESLIB
//         PROCLIB='$PRCLIB',            IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//INPUT   EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&PP25MB,DISP=(,PASS),
//           UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*----------------------------------------------------------------
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:        DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*----------------------------------------------------------------
//PTU923   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTU923,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
```

```
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//PAC7MB   DD DSN=&&PP25MB,DISP=(OLD,DELETE)
//PAC7PP   DD DSN=&OLDPP,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(100,20),RLSE),
//            DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//*-------------------------------------------------------------
//PAF900  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPAF900,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM),
//        COND=(0,NE,PTU923)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AN$BASE   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
```

```
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//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=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPACA15,PACFIM$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=((0,NE,PTU923),(0,NE,PAF900))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE   DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7DC$BASE  DD DUMMY
```

```
//B7D3$BASE  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
```

Retrieval of passwords (UTMP)

UTMP - Introduction

Principle

The purpose of this procedure is to retrieve the passwords of 2.N users, using as input the PE file, produced by the backup of users parameters (PARM procedure), to put 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 wrong creation, the procedure prints an error message in display format.

Result

The procedure integrates 2.0 or 2.5 users passwords in the Administration Database.

UTMP - Description of steps

2.N passwords retrieval: PTURMP

Code	Physical name	Type	Label
PAC7CE	&OLDPE	Input	Old version user parameters
BVP7GU	&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
//         INDSN='$INDSN',          INDEX OF SYSTEM NON VSAM FILES
//         STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//         OLDPE=,                              DS NAME OF OLD PE
//         PSBLIB='$PSBLIB',              LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',              LIBRARY OF DBD'S
//         RESLIB='$RESLIB',              IMS RESLIB
//         PROCLIB='$PRCLIB',             IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//PTURMP  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTURMP,GTURMP$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7GU   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
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
BVP7AE	&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
BVP8GR	&INDSV..BVPGR	Input	Administration Database data

Code	Physical name	Type	Label
BVP8GN	&INDSV..BVPGN	Input	Administration Database index
BVP7AE	&INDSV..BVPAE	Input	Error labels
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&&PACGZ	Input	Update transactions
PAC7MV	&&PAC7MV	Output	Transactions formatting (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
BVP8GR	&INDSV..BVPGR	Output	Administration Database Data file
BVP8GN	&INDSV..BVPGN	Output	Administration Database Index file
BVP8GY	&INDSV..BVPGY	Output	Administration Database extension
BVP8GJ	&INDSV..BVPGJ	Output	Administration Database journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index file
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data file
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS file of Development Database elements
B7D3$BASE	DUMMY	Input	DSMS index 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	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='$SCAT',         VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//         OUTL=$OUT,                 OUTPUT CLASS OF REPORTS
//         OLDUV=,                       DS NAME OF OLD UV
//         UWK=$UWK,                              WORK UNIT
//         SPAMB='(TRK,(100,10),RLSE)',     TRANSACTION SPACE
//         PSBLIB='$PSBLIB',            LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//         RESLIB='$RESLIB',            IMS RESLIB
//         PROCLIB='$PRCLIB',           IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*********************************************************
//INPUT  EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
```

```
//           DD DSN=$BCOB,DISP=SHR
//CARTE    DD DDNAME=SYSIN
//PAC7MB   DD DSN=&&UV25MB,DISP=(,PASS),
//           UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-------------------------------------------------------------
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:         DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//           DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//           DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-------------------------------------------------------------
//PTU922   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//           PARM=(DLI,BVPTU922,PTUREP$SUG,&BUF,
//           &SPIE&TEST&EXCPVR&RST,&PRLD,
//           &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//           &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//           DD DSN=&STEPLIB,DISP=SHR
//           DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//           DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:         DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   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=DFSRRC00,REGION=$REGSIZ,
//           PARM=(DLI,BVPAF900,PAAFIC$SUG,&BUF,
//           &SPIE&TEST&EXCPVR&RST,&PRLD,
//           &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//           &IRLM),
```

```
//         COND=(0,NE,PTU922)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//         DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME   DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//         SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV   DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//         SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW   DD DISP=(,DELETE),UNIT=&UWK,
//         SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX   DD DISP=(,DELETE),UNIT=&UWK,
//         SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY   DD DISP=(,DELETE),UNIT=&UWK,
//         SPACE=&SPAMB,
//         DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//*------------------------------------------------------------
//PACA15   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPACA15,PAAFIM$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM),
//         COND=((0,NE,PTU922),(0,NE,PAF900))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
```

```
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7DC$BASE  DD DUMMY
//B7D3$BASE  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
//*
```

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 Entities Occurrences, Relations, U-type Volumes are
not processed.

The P.I.A. calls are processed as the comments.

The P.I.A. types change between the 2.n and 3.n versions: to allow the transactions exchange between these 2 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 new version UPDT procedure and a revoked transactions file.

MB25 - Description of Steps

MB file retrieval: PTU926

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&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 VAPAC 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',             VA PAC SYSTEM VSAM CATALOG
//          STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//          OLDMB=,                          DS NAME OF OLD MB
//          UWK=$UWK,                                   WORK UNIT
//          PSBLIB='$PSBLIB',             LIBRARY OF PSB'S
//          DBDLIB='$DBDLIB',             LIBRARY OF DBD'S
//          RESLIB='$RESLIB',             IMS RESLIB
//          PROCLIB='$PRCLIB',            IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//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
//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(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*--------------------------------------------------------------
//PTU926  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTU926,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:        DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
```

```
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//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 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
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
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&OLDGY	Input	2.5 GY transactions
PAC7MV	&&PACGY	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=$BASE,              CODE OF VAPAC DATABASE
//         OUT=$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='$SCAT',        VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF LOAD-MODULES
//         OLDGY=,                        DS NAME OF OLD GY
//         UWK=$UWK,                               WORK UNIT
//         PSBLIB='$PSBLIB',              LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',              LIBRARY OF DBD'S
//         RESLIB='$RESLIB',              IMS RESLIB
//         PROCLIB='$PRCLIB',             IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//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
```

```
//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(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*------------------------------------------------------------
//PTU927  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTU927,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7GY   DD DSN=&OLDGY,DISP=SHR
//PAC7MV   DD DSN=&&PACGY,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

<u>Principle</u>

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
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&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=$BASE,                 CODE OF VAPAC DATABASE
//         OUT=$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='$SCAT',            VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',     LIBRARY OF LOAD-MODULES
//         OLDMB=,                            DS NAME OF OLD MB
//         UWK=$UWK,                                    WORK UNIT
//         PSBLIB='$PSBLIB',             LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',             LIBRARY OF DBD'S
//         RESLIB='$RESLIB',             IMS RESLIB
//         PROCLIB='$PRCLIB',            IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//VERIFY EXEC PGM=IDCAMS
//*--------------------
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*--------------------------------------------------------------
//PTU928  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTU928,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
```

```
//BVP7AE    DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN    DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR    DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU    DD DSN=&INDSV..BVPGU,DISP=SHR
//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
BVP7AE	&INDSV..BVPAE	Input	Error messages
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
PAC7GY	&OLDGY	Input	GY transactions of the new version
PAC7MV	&&PACGY	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=$BASE,              CODE OF VAPAC DATABASE
//         OUT=$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='$SCAT',         VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',    LIBRARY OF LOAD-MODULES
//         OLDGY=,                        DS NAME OF OLD GY
//         UWK=$UWK,                              WORK UNIT
//         PSBLIB='$PSBLIB',            LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',            LIBRARY OF DBD'S
//         RESLIB='$RESLIB',            IMS RESLIB
//         PROCLIB='$PRCLIB',           IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//*************************************************************
//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
//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(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*------------------------------------------------------------
//PTU929   EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTU929,PAFIC9$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//             DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//             BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//             BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7GY   DD DSN=&OLDGY,DISP=SHR
//PAC7MV   DD DSN=&&PACGY,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

Procedure	Comments
ARAD	Archiving Administration Database
ARPM	Archiving the QJ journal of SCM module

Procedure	Comments
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
INQJ	Initialization of SCM Tools Interface archive journal
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
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 journal 2.n
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	Extract line 'UNS' 3.0 utility
UTU2	Update line 'UNS' 3.0 utility'
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

Procedure	Comments
SIPM	Generation simulation
PPAD	PAF processor on Administration Database
UPPM	Update of elements
TRED	PacTransfer: print of parameters

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 than 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 3.0 version.

- If this version is installed in the same environment than 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,
  - One library for the DBDs,
  - One library for the PSBs.
- 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.

The DSNAME allocation is performed according to the following principles:

- the first index level(s) is/are represented by the symbolic parameters:
  - &INDSV for a system file with a VSAM organization,
  - &INDUV for a user file with a VSAM organization,
  - &INDSN for a system which organization is non-VSAM,
  - &INDUN for a user file which organization is non-VSAM or a generation file.

  The same value can be allocated easily to these different parameters.
- The last index level is represented by the name of the file. The length of the name is 5 characters for 'system' files and 6 characters for 'dynamic' files. The name format depends on the file type:
  - BVPxx for 'system' files,
  - &BASE.xx for 'dynamic' files,

with &BASE as symbolic parameter on four characters representing the VisualAge Pacbase database code, and xx on two characters characterizing the logical name of the file.

Two parameters are used to call the
- &SYSTCAT catalog made up of VSAM files belonging to the VisualAge Pacbase system,
- &VSAMCAT catalog made up of VSAM files belonging to the VisualAge Pacbase database.

The same value can be assigned easily to the two parameters.

These standards are used to enable several different VisualAge Pacbase systems to coexist on the same site, and several different databases to coexist in the same system.

Let us notice that such a configuration is unusual.

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 which is found in the batch load-module library (SBVPMBR8) with the name BVPSECUR.

Program	Security system
BVPSECRA	RACF
BVPTSS	TOPSECRET Batch

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 former sub-chapters, the product includes error labels and on-line documentation: AE.

Characteristics	Value
Organization	DL/1 HISAM database
Dsname	&INDSV..BVPAE
DBDName	BVPDAE
Lng SEGM	80 bytes

Characteristics	Value
Lng RECORD	80 bytes
Utilization	batch and on-line
Size	around 50,000 records for each language

JOB report

This database, which is dedicated to users of applications, stores all transactions issued from the 'GPRT' procedure (STEPS reports, compilation results) once the JOBS are executed in on-line mode via the 'JOB' command (with the 'SUB' command, this database is not taken into account). Each user can purge all or part of his/her JOBS reports (see chapter: 'CHOICE' in the User's Guide.) This action performed regularly avoids database overload.

Characteristics	Value
Organization	DL/1 HISAM/VSAM database
DSName	&INDSV..BVPLB
DBDName	BVPDLB
Lng SEGM	98 bytes
Lng RECORD	108 bytes
Size	38 records per CI of 4,096

Spas database

This database is used to transfer information from one VisualAge Pacbase transaction to another one. It simulates the function of SPA, and makes it possible to save the current dialog.

Characteristics	Value
Organization	DL/1 HISAM/VSAM database
DSName	&INDSV..BVPSV
DBDName	BVPDSV
Lng SEGM	6230 bytes
Lng RECORD	6240 bytes
Size	1 record per CI of 8,192

Generation Skeletons

The product also includes the following files:

- A skeleton generation 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 skeleton generation 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 skeleton generation 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 skeleton generation file (SN file) used by the eBusiness generator function.

Characteristics	Value
Size	Approx. 350 records
Organization	VSAM-KSDS
Recsize	4,605

Characteristics	Value
CI Size	5,120 (data) 1,536 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSN

- A skeleton generation 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 skeleton generation 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 skeleton generation 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 system utility programs parameters (they are implemented in the batch procedures used for the installation and run 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 at installation.

Caution

The information should not be modified except in particular cases.

Member	Contents or format	particular indication
DFBVPAE	DELETE/DEFINE AE	
DFBVPGN	DELETE/DEFINE GN	
DFBVPGS	DELETE/DEFINE GS	
DFBVPGU	DELETE/DEFINE GU	

Member	Contents or format	particular indication
DFBVPGK	DELETE/DEFINE GU	
DFBVPLB	DELETE/DEFINE LB	
DFBVPPA	DELETE/DEFINE PA	On-line PA file (1)
DFBVPP1	DELETE/DEFINE P1	
DFBVPSA	DELETE/DEFINE SA	
DFBVPSC	DELETE/DEFINE SC	
DFBVPSG	DELETE/DEFINE SG	
DFBVPSP	DELETE/DEFINE SP	
DFBVPSR	DELETE/DEFINE SR	
DFBVPSS	DELETE/DEFINE SS	
DFBVPSN	DELETE/DEFINE SN	
DFBVPWS	DELETE/DEFINE WS	
DFBVPSV	DELETE/DEFINE SV	
DFSYIANA	DEFINE	Modifiable
DLSYIANA	DELETE	
DFSYSEXT	DEFINE	Modifiable (2)
DLSYSEXT	DELETE	
DFSYTRDU	DEFINE	
DLSYTRDU	DELETE	
DFSYTRPF	DEFINE	
DLSYTRPF	DELETE	
DFWKREOR	DEFINE	
DLWKREOR	DELETE	
DFWKROAD	DEFINE	
DLWKROAD	DELETE	
DFWYREOR	DEFINE	
DLWYREOR	DELETE	
DFWYROAD	DEFINE	
DLWYROAD	DELETE	
DLBVPGJ	DELETE GJ	
DLBVPGR	DELETE GR	
DLBVPGY	DELETE GY	
DLBVPTR	DELETE TR	

Member	Contents or format	particular indication
DLBVPQJ	DELETE QJ	
DLPQCE	DELETE PQCE file	
LIBVPGJ	LISTCAT of GJ	
VERIFAE	VERIFY (PAC7AE)	
VERIFAN	VERIFY (PAC7AN)	
VERIFEM	VERIFY (PAC7EM)	
VERIFGN	VERIFY (PAC7GN)	
VERIFGU	VERIFY (PAC7GU)	
VERIFSC	VERIFY (PAC7SC)	
VERIFSG	VERIFY (PAC7SG)	
VERIFSP	VERIFY (PAC7SP)	
VERIFSR	VERIFY (PAC7SR)	
VERIFSS	VERIFY (PAC7SS)	
VERIFSN	VERIFY (PAC7SN)	
VERIFTD	VERIFY (PAC7TD)	
VERIFLB	VERIFY (PAC7LB)	
SRTPC25	SORT FIELDS	
SRTREO1	SORT FIELDS	
SRTREO2	SORT FIELDS	
LDBVPLB	LB maxi record	
LDBVPSV	SV maxi record	
LDBVPWS	WS maxi record	
LDBVPPA	PA maxi record	
MAXKEY	maxi record	
KEY01	record number 1	
REPRO	IDCAMS input	
REPROLB	IDCAMS LB input	
REPROSV	IDCAMS SV input	
REPROWS	IDCAMS WS input	
REPROPA	IDCAMS PA input	
REPRO999	IDCAMS input	
DFSVSAM8		
DFSVSAM9		

Member	Contents or format	particular indication
RANDOM	Randomization module source	
PACCTRL	to declare on IMS for on-line mode	

- (1) Size according to the number of on-line PAF requests sent when using the file.
- (2) Size according to the number of the inquiries from the deep extractor.

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
- GR database.

Characteristics	Value
Organization	DL/1 HDAM/OSAM database
DSName	&INDSV..BVPGR
DBDName	BVPDGR
Lng SEG	152
Utilization	Batch and on-line
Size	12 records per C.I. of 2,048

- The GY extension database.

Characteristics	Value
Organization	DL/1 HDAM/OSAM database

Characteristics	Value
DSName	&INDSV..BVPGY
DBDName	BVPDGY
Lng SEG	1026
Utilization	Batch and on-line
Size	?? record per C.I. of 2,048

- GN index database.

Characteristics	Value
Organization	DL/1 HISAM database
DSName	&INDSV..BVPGN
DBDName	BVPDGN
Lng SEG	59
Lng RECORD	68
Utilization	Batch and on-line

- Transactions database - GJ journal.

Characteristics	Value
Organization	DL/1 HDAM/OSAM database
DSName	&INDSV..BVPGJ
DBDName	BVPDGJ
Lng SEG	178
Utilization	Batch and on-line
Size	2 records per C.I. of 512

- GU user database.

Characteristics	Value
Organization	DL/1 HISAM database
DSName	&INDSV..BVPGU
DBDName	BVPDGU
Lng SEG	80
Lng RECORD	88
Utilization	Batch and on-line
Size	45 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.

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 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 databases contains all the data relating to the development of applications.

- AR database.

Characteristics	Value
Organization	DL/1 HDAM/OSAM database
DSName	&INDUV..&BASE.AR
DBDName	BDAR$BASE (batch) TDAR$BASE (on-line)
Lng SEG	152
Utilization	Batch and on-line
Size	12 record per C.I. of 2,048

- AY extension database.

Characteristics	Value
Organization	DL/1 HDAM/OSAM database
DSName	&INDUV..&BASE.AY
DBDName	BDAY$BASE (batch) TDAY$BASE (on-line)
Lng SEG	1,026
Utilization	Batch and on-line
Size	?? record per C.I. of 2,048

- AN index database.

Characteristics	Value
Organization	DL/1 HISAM database
DSName	&INDUV..&BASE.AN

Characteristics	Value
DBDName	BDAN$BASE (batch) TDAN$BASE (on-line)
Lng SEG	59
Lng RECORD	68
Utilization	Batch and on-line

- Transactions database - AJ journal.

All transactions in the database, in batch or on-line mode, are stored in order to restore the database if the system's standard security were to fail and to provide, on the other hand, the information source to the tool dedicated to the database evolution analysis.

Usually, these transactions are stored in the transactions back-up file (PJ), the transactions database is used only as an intermediary area from the time the transactions are processed by the system and the time they are transferred on their permanent media through the ARCH procedure.

Characteristics	Value
Organization	DL/1 HDAM/OSAM database
DSName	&INDUV..&BASE.AJ
DBDName	BDAJ$BASE (batch) TDAJ$BASE (on-line)
Lng SEG	178
Utilization	Batch and on-line
Size	2 records per C.I. of 512

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 SYSIN of the IDCAMS utility implemented in the batch procedures used for the working of the Development Databases.

Information relating to the file names and the disks (catalogue), 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: the VA Pac Database Manager can modify the size of the files that make up the Database, according to the changes it undergoes, in these members.

- 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 during 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)
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)
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: The system reads an additional scaleable database, either in batch mode, or in on-line mode, for DSMS module users (refer to the operations manual relating to the DSMS module). The database includes the list of

VisualAge Pacbase entities to update. They are sorted by change numbers. These change numbers are entered by the users on the first VisualAge Pacbase screen.

This database is completed with an Index database (primary index).

Note: These two databases must be defined and loaded via the relevant procedure (LDDC), even if the DSMS module is not used on site.

Characteristics	Value
Organization	DL/1 HIDAM/VSAM database
DSName	&DSMS..&BASE.DC
DBDName	BDDC$BASE (batch) TDDC$BASE (on-line)
Recsize	4,089
Lng SEGM	42 to 168 bytes
Key	31
Utilization	Batch and on-line

Characteristics	Value
Organization	DL/1 HIDAM/VSAM database
DSName	&DSMS..&BASE.D3
DBDName	BDD3$BASE (batch) TDD3$BASE (on-line)
Recsize	36 bytes
CI size	1,024
Key	31 bytes
Utilization	Batch and on-line

PAF: Two working databases are required to use the PAF module: one is used in batch mode and the other in on-line mode.
- The batch PA PAF working database

Characteristics	Value
Organization	DL/1 HIDAM/VSAM database
DSName	&INDUV..&BASE.PA
DBDName	BDPA$BASE
Lng SEGM	90 to 1,031 bytes
Lng RECORD	4,089 bytes
Size	Variable

- 'PA' Batch primary index database

Characteristics	Value
Organization	DL/1 INDEX/VSAM database
DSName	&INDUV..&BASE.P1
DBDName	BDP1$BASE
Lng SEGM	67 bytes
Lng RECORD	42 bytes
Size	24 records per CI of 1,024

.PAF/PUF: on-line PA working database

Characteristics	Value
Organization	DL/1 HIDAM/VSAM database
DSName	&INDSV..BVPPA
DBDName	BVPDPA
Lng SEGM	90 to 1,031 bytes
Lng RECORD	4,089 bytes
Size	Variable

- On-line 'PA' primary index database

Characteristics	Value
Organization	DL/1 INDEX/VSAM database
DSName	&INDSV..BVPP1
DBDName	BVPDP1
Lng SEGM	67 bytes
Lng RECORD	42 bytes
Size	24 records per CI of 1,024

- A schema extraction file GS containing the user's extractors and macro-commands. (PAF Module Extension)

Characteristics	Value
Organization	VSAM-KSDS
Recsize	230
CI size	4,096

Characteristics	Value
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

<u>Printed output</u>

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.

<u>Result</u>

Once the update is performed, the network is ready for either on line or batch use.

<u>Note</u>

Extracted transactions to be used by the ROAD procedure must be copied in a catalogued file by taking the following - otherwise comment - line into account:

`'//*VINS.PAC7MR  DD DSN=ROADFILE'.`

VING - Description of Steps

<u>Input recognition: PTU001</u>

<u>Initialization of the KSDS working file: IDCAMS</u>

<u>VSAM files check: IDCAMS</u>

<u>Update of the Administration Database: VING</u>

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error labels
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data
BVP7GU	&INDSV..BVPGU	Input	Administration Database Users
BVP8GJ	&INDSV..BVPGJ	Input	Administration Database Journal
BVP8GN	&INDSV..BVPGN	Input	Administration Database Index
BVP8GR	&INDSV..BVPGR	Input	Administration Database Data
BVP8GY	&INDSV..BVPGY	Input	Administration Database Extension

Code	Physical name	Type	Label
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 anomaly and restart the procedure
- 8 : No access authorization for batch procedure
- 12: Input-output error on a file

Deletion of the KSDS working file: IDCAMS

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=$OUT,              OUTPUT CLASS OF REPORTS
//         PSBLIB='$PSBLIB',       LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',       LIBRARY OF DBD'S
//         RESLIB='$RESLIB',       IMS RESLIB
//         PROCLIB='$PRCLIB',      IMS PROCLIB
//         FDIC=,                  DSN USER'S META-ENTITIES
//         USER=,                  USER CODE
//         UWK=$UWK,               WORK UNIT
//         SPAMB='(TRK,(5,1),RLSE)',   WORK FILE SPACE
//         SPAWK='(TRK,(50,10),RLSE)', WORK FILE SPACE
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM,BKO=
//*************************************************************
//INPUT    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&VINGMB,DISP=(,PASS),UNIT=&UWK,
//           SPACE=(TRK,(5,1),RLSE),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//CARTE    DD DDNAME=SYSIN
//*---------------------------------------------------------------
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER'
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$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
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*----------------------------------------------------------------
//VINS    EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPVINS,PAAFIM$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM,&BKO)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//         DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//         BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//BVP8GJ   DD DSN=&INDSV..BVPGJ,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP8GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP8GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//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)
//*-------------------------------------------------------------
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER',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.

<u>Execution conditions</u>

The Database must be closed to on-line processing.

<u>Abnormal execution</u>

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
		'R'	Retrieval of User Entities with the 'extension' format
		' '	'I' + 'R'

<u>Printed output</u>

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 conversion of the user entities into 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 in 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 several sessions.

Note

Extracted transactions to be used in input by the REOR procedure must be copied in a cataloged 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 working file: IDCAMS

VSAM files check: IDCAMS

Update of the Development Database : VINS

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error Labels
BVP7GN	&INDSV..BVPGN	Input	Administration Database Index
BVP7GR	&INDSV..BVPGR	Input	Administration Database Data
BVP7GU	&INDSV..BVPGU	Input	Administration Database Users
B7AJ$BASE	&INDUV..&BASE.AJ	Output	Development Database Journal
B7AN$BASE	&INDUV..&BASE.AN	Output	Development Database Index
B7AR$BASE	&INDUV..&BASE.AR	Output	Development Database Data

Code	Physical name	Type	Label
B7AY$BASE	&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	
SORTWK02		Sort	
SORTWK03		Sort	

Return codes:

- 0 : No error detected on files
- 4 : Correct the anomaly and restart the procedure
- 8 : No access authorization for batch procedure
- 12: Input-output error on a file

Deletion of the KSDS working 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=$BASE,         CODE OF VAPAC DATABASE
//         INDSV='$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
//         PSBLIB='$PSBLIB',        LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',        LIBRARY OF DBD'S
//         RESLIB='$RESLIB',        IMS RESLIB
//         PROCLIB='$PRCLIB',       IMS PROCLIB
//         FDIC=,                   DSN USER'S META-ENTITIES
//         USER=,                   USER CODE
//         UWK=$UWK,                WORK UNIT
//         SPAMB='(TRK,(5,1),RLSE)',   WORK FILE SPACE
//         SPAWK='(TRK,(50,10),RLSE)', WORK FILE SPACE
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM,BKO=
//****************************************************************-----*
//INPUT    EXEC PGM=BVPTU001
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//PAC7MB   DD DSN=&&VINSMB,DISP=(,PASS),UNIT=&UWK,
//            SPACE=(TRK,(5,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//CARTE    DD DDNAME=SYSIN
//*------------------------------------------------------------
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER'
```

```
//STEPLIB  DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
//         UNIT=&UWK,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-------------------------------------------------------------
//DEFINE EXEC PGM=IDCAMS
//*:STEPCAT  DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN    DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//*-------------------------------------------------------------
//MAXKEY  EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF   DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//MAXKEY   DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//*-------------------------------------------------------------
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//*-------------------------------------------------------------
//VINS    EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPVINS,PACFIM$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM,&BKO)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
```

```
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE  DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//B7AN$BASE  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE  DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//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=&&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)
//*-------------------------------------------------------------
//PRMSYS EXEC PGM=BVPRMSYS,PARM='&USER',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 'P' lines of programs containing the 'UNS' operator and without any information in the 'Level-Condition type' area. The user will have to verify the output file and delete in this file all the lines he 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 the result file name of lines to point in the startup JCL: NOMUT='... '

Result

A file containing UNS 'P' lines to point 'NOMUT'.

UTU1 - Input / Processing / Results

No user input.

UTU1 - Description of Steps

Extracting lines 'P' with 'UNS' operator : UTIUN1

Code	Physical name	Type	Label
PAC7PC	&INDUN..&BASE.PC(0)	Input	Sequential image of the Development Database
UTUTO	&NOMUT	Output	Lines 'P' with 'UNS' for UPDATE

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
//         DD DSN=$BCOB,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 set 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	Lines 'P' with 'UNS' for UPDATE
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 which 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' meta entities. These meta entities cannot be modified.

Once converted, the 'formatted' meta entities can be modified and enriched.

You will have to manually operate during this conversion which consists of several steps.

Conversion process

Step 1: Analysis of the Database relations (UTM1)

In 3.n version, a relation is associated with one entity type and not with several types as in 2.n version.

Any 'old' 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 of '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 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.

Step 2: Conversion of meta entities, relations and impacted entities and update (UTM2)

- The 'old' 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 suit 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 in display format and you will have to chose a new relation code.

  Any relation creation is performed in session 1 an in inter-library.

- 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 at 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. It may have many rejects because these entities created in 2.n version are much more controlled than in 3.n version. And manual interventions in the Database are required.

  The transactions are not journalized.

  The &UTM2MV transaction file is declared as permanent file to allow the user to visualize all the transactions concerned by the update.

Step 3: Inventory of fixtures

It is recommended to re-execute the first step to make sure that the Database does no more contain 'old' relations calls.

Otherwise, you will have to operate again and re-execute the following steps.

Step 4: Reorganization

When the conversion is suitable, a reorganization of the Database is required.

Execution conditions

None during the step 1 (UTM1).

For the step 2 (update), the AR, AN, AJ and AY files must be closed to online use (except for material allowing the Batch/Online concurrency).

Printed output

At the end of the step 1, a report is printed containing the list of 'old' relations calls.

At the end of the step 2 before the update, error messages are printed in display format.

At the end of the update, a report lists the encountered errors.

Result

Once the reorganization is performed, the result is a Database free from 'old' meta entities and 'old' relations calls.

UTM1 - Description of Steps

Analysis of the Database relations: PTUME1

Code	Physical name	Type	Label
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data
B7AN$BASE	&INDUV..&BASE.AN	Input	Development Database Index
PAC7ME	&UTM1FILE	Output	List of relations calls in all the entities (length = 221)
BVP7AE	&INDSV..BVPAE	Input	Error Labels
PAC7EQ		Report	Report

UTM1 - Execution JCL

```
//* ------------------------------------------------------
//*        VISUALAGE PACBASE
//*
//* ------------------------------------------------------
//*            - MIGRATION META-ENTITES OLD 1 -
//*
//* ------------------------------------------------------
//* ------------------------------------------------------
//BVPUTM1  PROC BASE=$BASE,            CODE OF VAPAC DATABASE
//         INDSV='$INDSV',             INDEX OF VSAM SYSTEM FILES
//         INDSN='$INDSN',           INDEX OF NON-VSAM SYSTEM FILES
//         INDUV='$INDUV',             INDEX OF VSAM USER FILES
//*:       VSAMCAT='$VCAT',          USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',          VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',  LIBRARY OF LOAD-MODULES
//         UWK=SYSDA,                 WORK UNIT
//         OUT=$OUT,                  OUTPUT CLASS
//         UTM1FILE=,                 OUTPUT FILE FOR UTM2
//         SPAME='(TRK,(300,10),RLSE)',  PC FILE SPACE
//         VOLS='SER=$VOLUN',         BACKUP VOLUME
//         UNITS=$UNITUN,             BACKUP UNIT
//         PSBLIB='$PSBLIB',          LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',          LIBRARY OF DBD'S
//         RESLIB='$RESLIB',          IMS RESLIB
//         PROCLIB='$PRCLIB',         IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//**************************************************************
//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
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//*------------------------------------------------------------
//PTUME1  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//        PARM=(DLI,BVPTUME1,PACFIC$SUG,&BUF,
//        &SPIE&TEST&EXCPVR&RST,&PRLD,
//        &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//        &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//           DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//           BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//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 in input of 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
B7AR$BASE	&INDUV..&BASE.AR	Input	Development Database data

Code	Physical name	Type	Label
B7AN$BASE	&INDUV..&BASE.AN	Input	Development Database index
PAC7ME	&UTM1FILE	Input	List of relations calls in all the entities
BVP7AE	&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
B7AR$BASE	&INDUV..&BASE.AR	Output	Development Database data
B7AN$BASE	&INDUV..&BASE.AN	Output	Development Database index
B7AY$BASE	&INDUV..&BASE.AY	Output	Development Database extension
B7AJ$BASE	DUMMY	Output	Development Database Journal
BVP7AE	&INDSV..BVPAE	Input	Error messages
BVP7GN	&INDSV..BVPGN	Input	Administration Database index
BVP7GR	&INDSV..BVPGR	Input	Administration Database data
BVP7GY	&INDSV..BVPGY	Input	Administration Database extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC$BASE	DUMMY	Input	DSMS elements file of the Development Database
B7D3$BASE	DUMMY	Input	Index of Development Database Elements
PAC7ME	DUMMY		Work file
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=$BASE,            CODE OF VAPAC DATABASE
//         INDSV='$INDSV',             INDEX OF VSAM SYSTEM FILES
//         INDSN='$INDSN',          INDEX OF NON-VSAM SYSTEM FILES
//         INDUV='$INDUV',             INDEX OF VSAM USER FILES
//*:       VSAMCAT='$VCAT',            USER VSAM CATALOG
//*:       SYSTCAT='$SCAT',            VA PAC SYSTEM VSAM CATALOG
//         STEPLIB='$HLQ..SBVPMBR8',   LIBRARY OF LOAD-MODULES
//         UWK=SYSDA,                  WORK UNIT
//         OUT=$OUT,                   OUTPUT CLASS
//         UTM1FILE=,                  OUTPUT FILE FOR UTM2
//         SPAMV='(TRK,(300,10),RLSE)',   PC FILE SPACE
//         VOLS='SER=$VOLUN',          BACKUP VOLUME
//         UNITS=$UNITUN,              BACKUP UNIT
//         PSBLIB='$PSBLIB',           LIBRARY OF PSB'S
//         DBDLIB='$DBDLIB',           LIBRARY OF DBD'S
//         RESLIB='$RESLIB',           IMS RESLIB
//         PROCLIB='$PRCLIB',          IMS PROCLIB
//         BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//         CKPTID=,MON=N,LOGA=0,FMTO=T,DBRC=$DBRC,IRLM=$IRLM
//****************************************************************
//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
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*----------------------------------------------------------------
//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)
//*-----------------------------------------------------------
//PTUME2  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPTUME2,PACFIC$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AR$BASE  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AN$BASE  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7MB   DD DSN=&&UTM2MB,DISP=(OLD,PASS)
//PAC7ME   DD DSN=&UTM1FILE,DISP=SHR
//PAC7MV   DD DSN=&&MV,DISP=(,PASS),
//         UNIT=&UNITS,VOL=&VOLS,
//         SPACE=&SPAMV,
//         DCB=(RECFM=FB,BLKSIZE=17000,LRECL=170)
//PAC7ET   DD SYSOUT=&OUT
//*-----------------------------------------------------------
//PACA15  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//         PARM=(DLI,BVPACA15,PACA15$SUG,&BUF,
//         &SPIE&TEST&EXCPVR&RST,&PRLD,
//         &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//         &IRLM,,BKO)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//         DD DSN=&STEPLIB,DISP=SHR
//         DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//         DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
```

```
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DSN=&&IMSLOG,DISP=(,PASS),UNIT=&UWK,
//            SPACE=&SPALG,
//            DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//            BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$BASE DD DUMMY
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//B7AY$BASE DD DSN=&INDUV..&BASE.AY,DISP=SHR
//BVP7GN   DD DSN=&INDSV..BVPGN,DISP=SHR
//BVP7GR   DD DSN=&INDSV..BVPGR,DISP=SHR
//BVP7GU   DD DSN=&INDSV..BVPGU,DISP=SHR
//BVP7GY   DD DSN=&INDSV..BVPGY,DISP=SHR
//B7D3$BASE DD DUMMY
//B7DC$BASE DD DUMMY
//PAC7IE   DD SYSOUT=&OUT
//PAC7IF   DD SYSOUT=&OUT
//PAC7MV   DD DSN=&&MV,DISP=(OLD,DELETE)
//PAC7ME   DD DUMMY
//PAC7RB   DD DUMMY
//PAC7RY   DD DUMMY
//*
//
```

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 components of the previous version and are available for the new installation.

Implementation:

- Define a MCS as following, replacing HBVPxxx by the FMID of the version to be deleted.
- Create and execute after parameterizing the following JCL which 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.        */.
```

IBM ®

Part Number: DEPIM001354A - 6805

Printed in USA