

VisualAge Pacbase



Installation Guide IMS/VS Server & Workstation Components

Version 3.0



VisualAge Pacbase



Installation Guide IMS/VS Server & Workstation Components

Version 3.0

Note

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

According to your licence agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:

<http://www.ibm.com/software/ad/vapacbase/productinfo.htm>

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

First Edition (September 2002)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.0

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

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

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

© **Copyright International Business Machines Corporation 1983,2002. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii	Loading of Error messages and on-line help.	44
Trademarks	ix	Initialization of specific files.	45
Chapter 1. Foreword	1	Installation of the Administration Database	46
Introduction	1	Users File Initialization	46
VisualAge Pacbase Architecture	1	Initialization of Generation Data Groups	46
Contents of Supply	2	Loading of the Administration Database	48
Bibliography	2	Initialization of the QJ archive file.	49
Chapter 2. Prerequisites	3	Loading of the Administration Model	50
Prerequisites for Server Environment	3	List of components with their date of creation	50
Hardware and Software	3	Installation of the Development Database	51
Disk Space	3	Allocation and Loading of Database Parameters	51
Prerequisites for Client Environment	3	Initialization of Generation Data Groups	55
Hardware	3	Loading of the Test Database	58
Disk Space	4	Loading of the Development Model	59
Software	4	Complement - Pac/Impact	60
Communication	4	Chapter 4. Installation/Re-installation of Client Components	65
Chapter 3. Installation of Server Environment	5	Things to Know Before Installing	65
Environment	5	Root Directory	65
Introduction	5	Installation Startup.	66
TP Environment	5	Fundamentals of VA Pac Client-Server Communication.	66
DL/1 databases organization.	5	Administrator & Developer workbench	69
Optimization of performance.	8	Open Jade and Tidy / Publishing facility	71
Environment Preparation	8	eBusiness Tools	72
Parameterization	8	VisualAge Pacbase WorkStation	73
Installation Preparation.	8	Pacbase Web Connection.	77
JCL Installation	11	Middleware	79
JCL List	14	Additional Information	81
JCL Parameters	18	How to Configure the IMS ITOC CPI-C Protocol	81
JCL Before/After Lines	20	Editing Communication Parameters	82
System Installation.	20	The bases.ini File	82
Allocation and Loading of System Parameters	20	The vaplocat.ini File	84
Initialization	33	Uninstalling Client Components	86
Renaming the Exit users batch programs	36	Chapter 5. Tests	87
Renaming the on-line PSBs	37	List of Utilities	87
Loading of the PSBs and DBDs sources	38	Installation Tests	87
Compilation of ACB	38	Generation-Print, TP and Batch Update Tests	87
Loading of Procedures	39	Administration Database Procedures Tests	88
Loading of Generation Skeleton Files.	42		

Development Database Procedures Tests	88	Retrieval of MB Transactions (MB30)	154
Extraction-Utility Tests	89	MB30 - Introduction	154
Chapter 6. Re-installation of Server	91	MB30 - Description of Steps	154
Chapter 7. Retrieval	95	MB30 - Execution JCL	154
Retrieval of VisualAge Pacbase 2.0 and 2.5	95	Retrieval of GY Transactions (GY30)	156
Operations to be Performed	95	GY30 - Introduction	156
Retrieval of User Parameters (PE25)	97	GY30 - Description of Steps	156
PE25 - Introduction	97	GY30 - Execution JCL	157
PE25 - Input / Processing / Results	97	Retrieval of journal file PJ (PJ25)	158
PE25 - Description of Steps	98	PJ25 - Introduction	158
PE25 - Execution JCL	100	PJ25 - Description of steps	159
Retrieval of the Development Database (PC25)	103	PJ25 - Execution JCL	160
PC25 - Introduction	103	Procedures - Summary Table of Changes	161
PC25 - Notes on Data Retrieval	104	Chapter 8. Components	165
PC25 - Input / Processing / Results	105	Server Environment Components	165
PC25 - Description of Steps	106	Introduction	165
PC25 - Execution JCL	110	On-Line Documentation	166
Generation-Print Commands Retrieval (PG20)	116	JOB report	166
PG20 - Introduction	116	Spas database	167
PG20 - Input / Processing / Results	117	Generation Skeletons	167
PG20 - Description of Steps	117	Parameters	169
PG20 - Execution JCL	121	Bibliothèque des paramètres systèmes	169
Generation-Print Commands Retrieval (PG25)	127	Batch Procedure JCL Libraries	172
PG25 - Introduction	127	Administration Database	172
PG25 - Input / Processing / Results	127	Administration Database Files	172
PG25 - Description of Steps	128	Administration Database Backup	174
PG25 - Execution JCL	131	Development Database	175
PEI Retrieval (PP25)	137	Development Database Files	175
PP25 - Introduction	137	Development Database Parameter Library	176
PP25 - Input / Processing / Results	137	Development Database Backup Files	177
PP25 - Description of Steps	138	Modules - Specific Files	178
PP25 - Execution JCL	139	Pac/Impact	178
Retrieval of Pac/Transfer Parameters (UV25)	143	DSMS	179
UV25 - Introduction	143	PAF	180
UV25 - Input / Processing / Results	143	Complementary Libraries and Files	182
UV25 - Description of Steps	144	Chapter 9. Appendix	183
UV25 - Execution JCL	146	Installation of the Administration Database	
Retrieval of MB Transactions (MB25)	149	Model	183
MB25 - Introduction	149	VING - Introduction	183
MB25 - Description of Steps	149	VING - Input / Processing / Results	183
MB25 - Execution JCL	150	VING - Description of Steps	184
Retrieval of GY Transactions (GY25)	151	VING - Execution JCL	185
GY25 - Introduction	151	Installation of the Development Database	
GY25 - Description of Steps	152	Model	189
GY25 - Execution JCL	152	VINS - Introduction	189
		VINS - Input / Processing / Results	189
		VINS - Description of Steps	190

VINS - Execution JCL 192 GS/HE/ZS Files Intialization Utilities 195

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 environment,
- Client environment,
- Communication.

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.

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:

- the system elements: programs, files (on-line help included), and parameters,
- the administrator's Database that contains user parameters and other parameters,

- one or more development Database(s).

THE CLIENT ENVIRONMENT

It consists of the following components:

- Administrator workbench,
- Developer workbench which includes the Batch and eBusiness modules (these modules can be installed independently),
- VisualAge Pacbase WorkStation,
- eBusiness tools,
- Pacbase Web Connection.

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 VA Pac servers,
- Workstation Components CD-Rom,
- VA Pac documentation CD-Rom.

Bibliography

Refer to the Administrator's Procedures manual for information on the procedures used by the Administrator in the following contexts:

- Databases management,
- Versions management,
- Management utilities.

For information on the management of user parameters, (update of access keys, user codes and access authorizations), refer to the Administrator workbench online help.

For information on the communication between the Security System possibly installed on site and VisualAge Pacbase (authentication controls), refer to the 'Security System Interfaces' manual.

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 approximatively the disk space necessary (in millions of bytes) to install the servers:

Disk space for the installation	Total	VSAM	Non VSAM
Total installation	200	110	90
Total System	140	75	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): 128 Mb (256 Mb advised).
- Software: Microsoft Windows Script Host (version 5.1 and onwards).

Disk Space

Required disk space:

- 58 Mb for the Administrator & Developer workbench.
- 15 Mb for the VisualAge Pacbase WorkStation.

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.

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 a IMS/VS, environment, the communication protocol must be: ITOC CPI-C.

Chapter 3. Installation of Server Environment

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 BVP980 sub-program found in the on-line programs library.

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

- ttttttt transaction code to connect to VisualAge Pacbase (Input Point),

This VA Pac TP transaction to be declared in IMS is transactional and use 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. It is also used for the extension (AY GY) and for the traces of PUF communication areas (TR).

The key to these files is a seven-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 DFSDHC10. 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: $X = \text{INT}((CI - 19) / (REC + 11))$ 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 $(2^{24} - 1)$) 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\$ROOT\$FILE.AN member found in the SY System 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 SStorage OPTimization, makes a multi-volume installation effective by outspreading the most frequently accessed data throughout the whole of the AR file.

The definition on several volumes of the AN 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

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.

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 (SMPDFCSI)
 - Initialization of SMP/E libraries (SMPICSI)
 - Allocation of SMP/E workfiles (SMPALLOC)
 - FMID assignment (SMPDEFZ)
 - Input creation: DDDEF technical files (SMPDDEF)
- Cartridge receipt
 - Definition of 'Target field' and 'distribution field' (BVPDDEF) files
 - Assignment of files 'Target field' and 'Distribution field' (BVPALLOC)
 - 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.SBVPIINST: 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,
- hlq.SBVVPF2: PC and PE initialization files,
- hlq.SBVPMBR8: batch load-modules,
- hlq.SBVPMTR8: on-line load-modules,
- hlq.SBVVPF5: SG SN SP SR SS skeletons,
- hlq.SBVVPF6: SC skeleton
- hlq.SBVVPF7: SF skeleton
- hlq.SBVVPF8: 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 in 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.

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 CHARACTERS)
===PRM CCPT=<>             .JOB ACCOUNTING CODES (JOB CARDS)
===PRM CLASSJ=1           .JOB EXECUTION CLASS (JOB CARDS)
===PRM MSGCL=A            .JCL OUTPUT CLASS (MSGCLASS)
===PRM INDSV='EXP.BVP3V'   .SYSTEM VSAM FILES INDEX
===PRM INDSN='EXP.BVP3N'   .SYSTEM NON VSAM FILES INDEX
===PRM INDUV='UTI.BVP3V'   .USER VSAM FILES INDEX
===PRM INDUN='UTI.BVP3N'   .USER NON VSAM FILES INDEX
===PRM BASE='BASE'        .NAME OF DATABASE DEVELOPMENT (4 CHAR)
===PRM OUT=H              .JOB SYSOUT CLASS
===PRM UWK=SYSDA          .WORK FILES UNIT
===PRM UNITSN=SYSDA       .NON VSAM SYSTEM FILES UNIT
===PRM UNITUN=SYSDA       .NON VSAM USER FILES UNIT
===PRM VOLSN=             .SYSTEM NON VSAM FILES VOL=SER=
===PRM VOLSV=             .SYSTEM VSAM FILES VOLUME
===PRM VOLUN=             .USER NON VSAM FILES VOL=SER
===PRM VOLUV=             .USER VSAM FILES VOLUME
===PRM VCAT=              .USER FILES VSAM CATALOG
===PRM SCAT=              .SYSTEM FILES VSAM CATALOG
===PRM LSK='A'            .LANGAGE OF SKELETON FILES
===PRM DSCB='BVP.DSCB'    .DSCB MODEL FILE DSNAME
===PRM BIBP='SYS1.PROCLIB' .PROCEDURE LIBRARY
===PRM HLQ='HLQ'          .HEAD LEVEL QUALIFIER OF LOAD MODULES
===PRM BIBT='SYS1.SORTLIB' .SORT LIBRARY
===PRM BCOB='SYS1.SCEERUN' .COBOL ROUTINE LIBRARY
===PRM CSDL='CICS410.SDFHLOAD' .DFHCSDUP STEPLIB DSN
===PRM DFHCSD='BVP.DFHCSD' .CSD DSN
===PRM GROUF='BVPGR'       .'GROUP' NAME FOR FILE
===PRM GROUP='BVPGR'       .'GROUP' NAME FOR PROGRAM
===PRM GROUT='BVPGR'       .'GROUP' NAME FOR TRANSACTION
===PRM LIST=<>             .'LIST' NAME
===PRM TABTDF=<>           .DSN OF TABLES DESCRIPTION FILE
===PRM DSMS=NULLFILE       .DSN OF PRODUCT ELEMENTS (DC)
===PRM CSOC='DBSS'         .TRANSACTION CODE "SOCKET" (4 CAR)
===PRM CPIC='DBST'         .TRANSACTION CODE "CPIC" (4 CAR)
===PRM TRPM='DBPM'         .TRANSACTION AUTOMAT PUF (4 CAR)
===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
===SELM	mmmm1 mmmm2 ... (1)	Selection of JCL modules
===PRM	PPPP=pppp (2)	Parameter
===BEGMOD		Insertion of lines at the beginning of module
....1		Lines to be added before each module
....n		
===ENDMOD		Insertion of lines at the end of module
....1		Lines to be added after each module
....n		

(1) mmmm1 = name of JCL module

mmmm2 = name of JCL module...

(2) 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

No particular module is selected. They are all 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: IEUBUPDTE.

NOTE: Because of this default option, all './' characters found in JCL modules containing IEUBUPDTE were replaced with ':/'.

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

REPORT

MM1JCL 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

TABLE OF INSTALLATION JCLS

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 load modules batch Exit users	
D04PSREN	Naming of PSBs TP	
D04SRCD	Loading of DBDs sources	

Member	Contents	Procedure
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
D08TINQJ	Initialization of archival journal file VisualAge 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	
I03GDG	Definition of PC and PJ files' DSCBs	
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
	STANDARD PROCEDURES	
JLACTI	Activity follow-up	ACTI
JCLARAD	Archival of the administration database journal	ARAD
JCLARCH	Archival of the development database journal	ARCH
JCLCPSN	Comparison between sub-networks	PACX

Member	Contents	Procedure
JCLCSES	Compression of session number	CSES
JCLESES	Matching session numbers	ESES
JCLEMLD	Loading of error messages related to generated applications	EMLD
JCLEMUP	Update of error messages related to generated applications	EMUP
JCLEXLI	Extraction of libraries	PACX
JCLEXPJ	Extraction of transactions from journal	PACX
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
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
JCLROAD	Reorganization of administration Database	ROAD
JCLRSAD	Backup of administration Database	RSAD
JCLSADM	SSADM method integrity control	SADM
JCLSASN	Backup of sub-networks	PACS
JCLSASY	Complement system backup	SASY
JCLSAVE	Backup of development Database	PACS
JCLSTOP	Optimization of multi-number data	STOP
JCLSVAD	Backup of administration Database	PACG
JCLUPDP	Update of PAF batch format	UPDP

Member	Contents	Procedure
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
	STANDARD BRIDGE MODULE	
JCLARPM	Archival of Standard bridge journal	ARPM
JCLCHPM	Integrity Control of Events/Elements	CHPM
JCLCPPM	Comparision with extracted files	CPPM
JCLEXPM	Extraction of Database Data	EXPM
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
	PAC/TRANSFER 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

Member	Contents	Procedure
JCLISOS	Selection of strings, operators	ISOS
	RETRIEVAL OF FORMER RELEASES	
RGY250	Retrieval of GY file from the 2.5	GY25
RMB250	Retrieval of MB file from the 2.5	MB25
RGY300	Retrieval of GY file from 3.0 to 2.5	GY30
RMB300	Retrieval of MB file from 3.0 to 2.5	MB30
RPC250	Retrieval of PC file from the 2.5	PC25
RPE250	Retrieval of PE file from the 2.5	PE25
RPG200	Retrieval of PG file from the 2.0	PG20
RPG250	Retrieval of PG file from the 2.5	PG25
RPP250	PEI retrieval from 2.5 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

Parameter	Meaning	Default
CCPT	Job accounting code	<>
CLASSJ	Job execution class	1
MSGCL	JCL output class	A
INDSV	VSAM system (maximum: 24 characters)	'EXP.BVP3V'
INDSN	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)	BASE
OUT	SYSOUT print class	H
UWK	UNIT of work files used	SYSDA
UNITSN	UNIT of non-VSAM system files	SYSDA
UNITUN	UNIT of NON VSAM user files	SYSDA
VOLSN	VOL=SER of NON-VSAM files	
VOLSV	VOL=SER of VSAM system files	
VOLUN	VOL=SER of NON-VSAM user file	
VOLUV	VOL=SER of VSAM user files	
VCAT	VSAM catalog of the development Database (user files)	
SCAT	VSAM system catalog (System files)	
LSK	Skeleton language	A
BIBP	DSNAME of the procedure library	'SYS1.PROCLIB'
HLQ	Prefix of the batch and on-line load-module library	'HLQ'
BIBT	DSNAME of sort library (SORTLIB)	'SYS1.SORTLIB'
BCOB	DSNAME of COBOL routine library	'SYS1.SCEERUN'
	Update of CICS CDS	
TABTDF	Table description file DSN (1)	<>
	DSMS module	
DSMS	Development database-element file name (1)	<>
REGSIZ	Region size for batch procedures	1536K

Parameter	Meaning	Default
IMSID	IMSID parameter for batch procedures (N=NO)	N
IRLM	Use of IRLM in batch procedurs (N=NO)	N
DBRC	Use of DBRC in batch procedures (N=NO)	N
SUG	VA Pac PSBs suffix (2 characters)	30
DBDLIB	VA Pac DBDs library	IMSVS.DBDLIB
PSBLIB	VA Pac PSBs library	IMSVS.PSBLIB
ACBLIB	ACB library	IMSVS.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.

JCL Before/After Lines

LINES BEFORE AND AFTER JCL MODULES

===BEGMOD1)) Lines to be inserted before each JCL modulen)

===ENDMOD1)) Lines to be inserted after each JCL modulen)

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 BVPMLLCL utility into as many members as there are JCL modules.

This utility adds1 ton lines in front of each JCL module and1 ton 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

Step	Program	Comments
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.D2C' JOB

Loading of \$INDSN..BVPSY file

Step	Program	Comments
STEP1	IEBUPDTE	Loading of PDS members

CAUTION

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

EXECUTION JCL

```

//$PRFJ.SY JOB ($CCPT),'PDS-PARAM.',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 ADDRESS- *
*          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 SOC7 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	
EPA1	DC	CL20'&SYSDATE &SYSTIME'	
	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	TRANSFERT CLE (R9) DS ZONE TAMPON
	LR	R4,R1	SAVE ADRESSE DMBDACS AVANT TRT
	TRT	TMPDECB(1),TBALPHA	LE 1ER CARACTERE EST-IL ALPHA ?
	BNZ	SUITE	NON : (ON SE DEBRANCHE)
	MVC	PSTDECB(7),TMPDECB	OUI : DECALAGE DE LA ZONE TAMPON
	MVC	TMPDECB+1(7),PSTDECB	DE 1 OCTET VERS LA DROITE
	TR	TMPDECB(1),TBNUM1	ECRASEMNT 1ER OCTET ALPHA -> NUM
	TR	TMPDECB+1(1),TBNUM2	ECRASEMNT 2ME OCTET ALPHA -> NUM
SUITE	LA	R5,1(R5)	LA CLE FAIT DONC 1 OCTET DE PLUS
	LR	R1,R4	RESTAURATION ADRESSE DMBDACS
	EX	R5,TRT	LA CLE EST-ELLE NUMERIQUE
	BNZ	NONUM	NON : (DEBRANCHEMENT)
	XC	PSTDECB(8),PSTDECB	INIT FOR CVB
	EX	R5,PACK	ON PACK LA ZONE TAMPON DS PSTDECB
	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	REMISE A ZERO R4 (SECURITE)
	LH	R6,DMBDARAP	LOAD NOMB. SEGMENT PAR CI
	DR	R4,R6	R4 =N0 DE RAP DANS BLOCK
*			R5 =N0 CI CORRESPONDANT SANS
*			TENIR COMPTE DES BIT MAPS
	LA	R4,1(R4)	+1 POUR NE PAS COMMENCER A 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',DMBPF0BK	LONGUEUR DU CI EN OCTETS
	LTR	R8,R8	OSAM ?
	BNZ	OKBLK	YES
OKBLK	ICM	R8,B'0011',DMBPF0BK	LONGUEUR DU CI EN OCTETS
	DS	0H	
	LR	R6,R8	
	DROP	R6	
	LH	R2,DMBDARAP	NB DE POINTS D'ANCRAGE
	LA	R2,1(R2)	+1 POUR FSEAP
	SLL	R2,2	NB D'OCTETS DE CTRL BIT MAP
	SR	R6,R2	NB D'OCTETS UTILES BIT MAP
	SLL	R6,3	NB DE BITS UTILES BIT MAP
	BCTR	R6,R0	NB DE CI DECRITS PAR LA BIT MAP (SANS COMPTER LA BIT MAP)
*			
	SR	R2,R2	
	DR	R2,R6	R3 = NB BIT MAPS AVANT -1.
	AR	R5,R3	N0 CI CORRESPONDANT -1.
	LA	R5,2(R5)	N0 CI CORRESPONDANT (+1 POUR LA BIT MAP +1 CAR LE 1ER CI EST LE CI N0 1)
*			
*			
	SLL	R5,8	FORMAT BBB0
	OR	R4,R5	FORMAT BBBS
	B	FIN	
NONUM	DS	0H	
	LR	R1,R4	RESTAURATION ADRESSE DMBDACS
	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 ($VCAT) */
:/      ADD NAME=DFBVPSPG
DELETE ($INDSV..BVPSPG) CLUSTER
DEFINE CLUSTER ( NAME ($INDSV..BVPSPG)     -
                SHR (2,3) RUS KEY (5 0)    -
                VOL ($VOLSV)               -
                RECSZ (4605,4605) )        -
INDEX   ( NAME ($INDSV..BVPSPG.I)         -
        CISZ (256) )                       -
DATA    ( NAME ($INDSV..BVPSPG.D)         -
        FSPC (10,5) SPEED CYL (3 1)      -
        CISZ (5120) ) /*: CATALOG ($SCAT) */
:/      ADD NAME=DFBVPSPN
DELETE ($INDSV..BVPSPN) CLUSTER
DEFINE CLUSTER ( NAME ($INDSV..BVPSPN)     -
                SHR (2,3) RUS KEY (5 0)    -
                VOL ($VOLSV)               -
                RECSZ (4605,4605) )        -
INDEX   ( NAME ($INDSV..BVPSPN.I)         -
        CISZ (256) )                       -
DATA    ( NAME ($INDSV..BVPSPN.D)         -
        FSPC (10,5) SPEED CYL (3 1)      -
        CISZ (5120) ) /*: CATALOG ($SCAT) */
:/      ADD NAME=DFBVPSPS
DELETE ($INDSV..BVPSPS) CLUSTER
DEFINE CLUSTER ( NAME ($INDSV..BVPSPS)     -
                SHR (2,3) RUS KEY (5 0)    -
                VOL ($VOLSV)               -
                RECSZ (4605,4605) )        -
INDEX   ( NAME ($INDSV..BVPSPS.I)         -
        CISZ (256) )                       -
DATA    ( NAME ($INDSV..BVPSPS.D)         -
        FSPC (10,5) SPEED CYL (5 1)      -
        CISZ (5120) ) /*: CATALOG ($SCAT) */
:/      ADD NAME=DFBVPSP
DELETE ($INDSV..BVPSP) 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 (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 ($VCAT) */
:/
ADD NAME=DFBVPCS
DELETE ($INDSV..BVPCS) CLUSTER
DEFINE CLUSTER ( NAME ($INDSV..BVPCS)           -
                SHR (2,3) RUS   KEY (4 0)      -
                VOL ($VOLSV)                   -
                RECSZ (3204,3204) )            -
INDEX   ( NAME ($INDSV..BVPCS.I)              -
        CISZ (256) )                          -
DATA    ( NAME ($INDSV..BVPCS.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 ($VCAT) */
:/      ADD NAME=DFSYSXEXT
DELETE ($INDUV..SYSXEXT.&USER) CLUSTER
SET LASTCC = 0
SET MAXCC = 0
DEFINE CLUSTER ( NAME ($INDUV..SYSXEXT.&USER)      -
                SHR (2 3) RUS KEYS(43 0)          -
                VOL ($VOLUV)                       -
                CYL (5 1)                           -
                RECSZ (254 254) )                  -
INDEX   ( NAME ($INDUV..SYSXEXT.&USER.I)          -
        CISZ (512) )                               -
DATA   ( NAME ($INDUV..SYSXEXT.&USER.D)          -
        FSPC (10 5) SPEED                         -
        CISZ (4096) )                               /*: CATALOG ($VCAT) */
:/      ADD NAME=DLSYSXEXT
DELETE ($INDUV..SYSXEXT.&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..WKREOR) CLUSTER
SET LASTCC= 0
SET MAXCC= 0
DEFINE CLUSTER ( NAME ($INDUV..WKREOR) -
        SHR (2 3) RUS KEYS(8 0) -
        VOL ($VOLUV) -
        CYL (1 1) -
        RECSZ (162 162) ) -
INDEX ( NAME ($INDUV..WKREOR.I) -
        CISZ (512) ) -
DATA ( NAME ($INDUV..WKREOR.D) -
        FSPC (10 5) SPEED -
        CISZ (4096) ) /*: CATALOG ($VCAT) */
:/ ADD NAME=DFWYREOR
DELETE ($INDUV..WYREOR) CLUSTER
SET LASTCC= 0
SET MAXCC= 0
DEFINE CLUSTER ( NAME ($INDUV..WYREOR) -
        SHR (2 3) RUS KEYS(8 0) -
        VOL ($VOLUV) -
        CYL (1 1) -
        RECSZ (1028 1028) ) -
INDEX ( NAME ($INDUV..WYREOR.I) -
        CISZ (512) ) -
DATA ( NAME ($INDUV..WYREOR.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=DLBVPQJ
DELETE ($INDSV..BVPQJ) NONVSAM
:/      ADD NAME=DLBVPTR
DELETE ($INDSV..BVPTR) NONVSAM
:/      ADD NAME=VERIFTD
VERIFY FILE (PAC7TD)
:/      ADD NAME=VERIFAN
VERIFY FILE (PAC7AN)
:/      ADD NAME=VERIFAJ
VERIFY FILE (PAC7AJ)
:/      ADD NAME=VERIFAE
VERIFY FILE (PAC7AE)
:/      ADD NAME=VERIFAR
VERIFY FILE (PAC7AR)
:/      ADD NAME=VERIFAY
VERIFY FILE (PAC7AY)
:/      ADD NAME=VERIFEM
VERIFY FILE (PAC7EM)
:/      ADD NAME=VERIFGJ
VERIFY FILE (PACGGJ)
:/      ADD NAME=VERIFGN
VERIFY FILE (PACGGN)
:/      ADD NAME=VERIFGR
VERIFY FILE (PACGGR)
:/      ADD NAME=VERIFGU
VERIFY FILE (PACGGU)
:/      ADD NAME=VERIFGY
VERIFY FILE (PACGGY)
:/      ADD NAME=VERIFQJ
VERIFY FILE (PAC7QJ)
:/      ADD NAME=VERIFSC
VERIFY FILE (PAC7SC)
:/      ADD NAME=VERIFSG
VERIFY FILE (PAC7SG)
:/      ADD NAME=VERIFSN
VERIFY FILE (PAC7SN)
:/      ADD NAME=VERIFSS
VERIFY FILE (PAC7SS)

```



```

//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
//*
//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: Job '\$prfj.D3G'

Allocation model DSCB - Definition GDG

Step	Program	Comments
STEP1	IDCAMS	Delete DSCB PE PK
STEP2	IEFBR14	Allocation of model DSCB

EXECUTION JCL

```

//$PRFJ.D3G JOB ($CCPT), 'PREPARATION', CLASS=$CLASSJ,
// MSGCLASS=$MSGCL
//*****
//* VISUALAGE PACBASE *
//* * *
//* INSTALLATION : DEFINITION DSCB *
//* * *
//*****
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$OUT
//SYSIN DD *
        DELETE ($INSDN..DSCB.BVPPE)
        DELETE ($INSDN..DSCB.BVPPK)
//*
//STEP2 EXEC PGM=IEFBR14
//DSCBPE DD DSN=$INSDN..DSCB.BVPPE, DISP=(,CATLG,DELETE),
//        SPACE=(TRK,(0)),VOL=SER=$VOLSN,UNIT=$UNITSN,
//        DCB=(RECFM=VB,LRECL=1023,BLKSIZE=27998)
//DSCBPK DD DSN=$INSDN..DSCB.BVPPK, DISP=(,CATLG,DELETE),
//        SPACE=(TRK,(0)),VOL=SER=$VOLSN,UNIT=$UNITSN,
//        DCB=(RECFM=FB,LRECL=170,BLKSIZE=6800)
//*

```

D03INI module: Job '\$prfj.D3I'

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" *

```

```

//*****
//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)
//*

```

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
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=((BVPRD00B,BVPRD000))
            S M=((BVPRIT0B,BVPRIT00))
            S M=((BVPRLGFB,BVPRLGFB))
            S M=((BVPRLGFB,BVPRLGFB))
            S M=((BVPRMCOB,BVPRMCO0))
            S M=((BVPRPARB,BVPRPAR))
            S M=((BVPRPRCB,BVPRPRC))
            S M=((BVPRPVPB,BVPRPVP))
            S M=((BVPRSECB,BVPRSEC))
            S M=((BVPRTFRB,BVPRTRF))

```



```

        S M=((BVPRUSEB,BVPRUSE))
        S M=((BVPRVL0B,BVPRVL00))
        S M=((BVPTPSTB,BVPTPST))
        S M=((BVPTPWSB,BVPTPWS))

/*
/**
//STEP2   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//I1      DD DSN=&&TEMP,DISP=(OLD,DELETE)
//O1      DD DSN=$HLQ..SBVPMBR8,DISP=SHR
//SYSIN   DD *
        C   I=((I1,R)),0=01
/*

```

Renaming the on-line PSBs

D04PSREN module: Job '\$prfj.D4P'

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
//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=01
        S M=((BVPRIMST,BVPRIMS))
        S M=((BVPSCPIT,BVPSCPI))
        S M=((BVPSSOCT,BVPSSOC))
/*
/**
//STEP2   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=$OUT
//I1      DD DSN=&&TEMP,DISP=(OLD,DELETE)
//O1      DD DSN=$HLQ..SBVPPSB,DISP=SHR
//SYSIN   DD *
        C   I=((I1,R)),0=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:

- 30 for the \$SUG parameter,
- P300 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 SCPI value for yyyy.

This is the PSB, which is used for the CPIC 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: P300.

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:

- BDxxxxyy
- TDxxxxyy

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

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=($PRCLIB)
//ET010 EXEC ACBGEN, SOUT='SOUT'
//*****
//*                               COMPILATION OF ACB                               *
//*                               -----                               *
//* THIS JOB MUST BE EXECUTED AFTER LOADING DBDLIB                               *
//* AND PSBLIB (EITHER BY LOADING OBJECT MODULES                               *
//* OR BY COMPILING DBD AND PSB).                                               *
//*****
//G.SYSIN DD *
BUILD DBD=(BVPDGR,BVPDGN,BVPDGR,BVPDGR,BVPDGR)
BUILD DBD=(BVPDAE,BVPDSV,BVPDLB,BVPDWS,BVPDTR)
BUILD DBD=(BVPDPA,BVPDP1)
BUILD DBD=(TDAN$BASE,TDAR$BASE,TDAJ$BASE,TDAY$BASE)
BUILD DBD=(BDAN$BASE,BDAR$BASE,BDAJ$BASE,BDAY$BASE)
BUILD DBD=(TDDC$BASE,TDD3$BASE)
BUILD DBD=(BDDC$BASE,BDD3$BASE)
BUILD DBD=(TDPA$BASE,TDP1$BASE)
BUILD DBD=(BDPA$BASE,BDP1$BASE)
BUILD PSB=(BVPRIMS,BVPSMPI)
BUILD PSB=(PACB$SUG,PACQ$SUG)
//G.IMSACB DD DSN=$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=BVPARAD
:/ ADD NAME=BVPARCH
:/ ADD NAME=BVPARPM
:/ ADD NAME=BVPGPRP
:/ ADD NAME=BVPGPRT
:/ ADD NAME=BVPGPRU
:/ ADD NAME=BVPUPDT
:/ ADD NAME=BVPUPDP
:/ ADD NAME=BVPPACG
:/ ADD NAME=BVPPACS
:/ ADD NAME=BVPPACX
:/ ADD NAME=BVPREOR
:/ ADD NAME=BVPROAD
:/ ADD NAME=BVPREST
:/ ADD NAME=BVPRSAD
:/ ADD NAME=BVPRESY
:/ ADD NAME=BVPSASY
:/ ADD NAME=BVPVINS
:/ ADD NAME=BVPVING
:/ ADD NAME=BVPPPAF
:/ ADD NAME=BVPINAE
:/ ADD NAME=BVPINGU
:/ ADD NAME=BVPINQJ
:/ ADD NAME=BVPSADM
:/ ADD NAME=BVPYSMC
:/ ADD NAME=BVPSTOP
:/ ADD NAME=BVPEMLD
:/ ADD NAME=BVPEMUP
:/ ADD NAME=BVPGETA
:/ ADD NAME=BVPGETD
:/ ADD NAME=BVPGETI
:/ ADD NAME=BVPSMTD
:/ ADD NAME=BVPRMTD
:/ ADD NAME=BVPACTI
:/ ADD NAME=BVPPQCA
:/ ADD NAME=BVPPQCE
:/ ADD NAME=BVPHIPM
:/ ADD NAME=BVPSIPM
:/ ADD NAME=BVPEXPM
:/ ADD NAME=BVPCPPM
:/ ADD NAME=BVPCHPM
:/ ADD NAME=BVPTRED
:/ ADD NAME=BVPTRDU
:/ ADD NAME=BVPTRJC
:/ ADD NAME=BVPTRPF
:/ ADD NAME=BVPTRRP
:/ ADD NAME=BVPTRUP
:/ ADD NAME=BVPIANA
:/ ADD NAME=BVPIGRA
:/ ADD NAME=BVPIMFH
:/ ADD NAME=BVPINFP
:/ ADD NAME=BVPINFQ
:/ ADD NAME=BVPIPEP

```

```

:/ ADD NAME=BVPIPFQ
:/ ADD NAME=BVPIPIA
:/ ADD NAME=BVPISEP
:/ ADD NAME=BVPI SOS
:/ ADD NAME=BVPPRGS
:/ ADD NAME=BVPXPAF
:/ ADD NAME=BVPXPDM
:/ ADD NAME=BVPINSL
:/ ADD NAME=BVPCSES
:/ ADD NAME=BVPESES
:/ ADD NAME=BVPGY25
:/ ADD NAME=BVPGY30
:/ ADD NAME=BVPMB25
:/ ADD NAME=BVPMB30
:/ ADD NAME=BVPPC25
:/ ADD NAME=BVPPE25
:/ ADD NAME=BVPPG20
:/ ADD NAME=BVPPG25
:/ ADD NAME=BVPPJ25
:/ ADD NAME=BVPPP25
:/ ADD NAME=BVPUV25
:/ ADD NAME=BVPSTAT
:/ ADD NAME=BVPUTAG
:/ ADD NAME=BVPUTFG
:/ ADD NAME=BVPUTSD
:/ ADD NAME=BVPUTU1
:/ ADD NAME=BVPUTU2
:/ ADD NAME=BVPLDLB
:/ ADD NAME=BVPLDPA
:/ ADD NAME=BVPLDP1
:/ ADD NAME=BVPLDDC
:/ ADD NAME=BVPLDSV
:/ ADD NAME=BVPLDWS
F+
//

```

Loading of Generation Skeleton Files

D06SKEL MODULE: '\$prfj.D6' JOB

Creation and loading of skeleton files.

Step	Program	Comments
STEP1	IDCAMS	DELETE/DEFINE SC SG SN SR SS SP
STEP2	IDCAMS	loading (REPRO) of 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 SC SG SN SR SS SP *
//* .STEP2 : LOADING VSAM SKELETON FILES 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=$INDSN..BVPSY(DFBVPCS),DISP=SHR
// DD DSN=$INDSN..BVPSY(DFBVPSG),DISP=SHR
// DD DSN=$INDSN..BVPSY(DFBVPSN),DISP=SHR
// DD DSN=$INDSN..BVPSY(DFBVPSR),DISP=SHR
// DD DSN=$INDSN..BVPSY(DFBVPS),DISP=SHR
// DD DSN=$INDSN..BVPSY(DFBVPS),DISP=SHR
//*
//STEP2 EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=$SCAT,DISP=SHR
//SYSPRINT DD SYSOUT=$OUT
//SCO DD DSN=$INDSV..BVPSC,DISP=SHR
//SGO DD DSN=$INDSV..BVPSC,DISP=SHR
//SNO DD DSN=$INDSV..BVPSN,DISP=SHR
//SRO DD DSN=$INDSV..BVPSR,DISP=SHR
//SSO DD DSN=$INDSV..BVPS,DISP=SHR
//SPO DD DSN=$INDSV..BVPS,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=$HLQ..SBVPF5(BVPSP),DISP=SHR
//SYSIN DD *
REPRO INFILE (SCI) OUTFILE (SCO)
REPRO INFILE (SGI) OUTFILE (SGO)
REPRO INFILE (SNI) OUTFILE (SNO)
REPRO INFILE (SRI) OUTFILE (SRO)
REPRO INFILE (SSI) OUTFILE (SSO)
REPRO INFILE (SPI) OUTFILE (SPO)
//*
//STEP3 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=$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: Job '\$prfj.D7'

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.

o 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 BVPIAE
$$LSK
**
/*
```

Initialization of specific files

D07INIT module: Job '\$prfj.I7'

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.

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.

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 backup.

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

Step	Program	Comments
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 that is older than the 3.0 version, consult the chapter dedicated to the retrieval of user parameters.

CAUTION

To use VisualAge Pacbase, you need an access key. You must enter here the key that you were given at delivery.

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 LANGAGE CODE OF THE SITE ON Y LINE COL 10-11 *
//* COMPLETE THE SYSTEM ACCESS KEY AS REQUESTED *
//*****
//TESTRSAD EXEC BVPRSAD
*ADMIN ADMIN
Y04000 EN
C<ACCES KEY (70 CHARACTERS)>
//
```

Initialization of the QJ archive file

D08TINQJ MODULE: '\$prfj.D8T' JOB

This JCL should be executed only at the first installation.

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

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

EXECUTION JCL

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

Loading of the Administration Model

D08XMET MODULE: '\$prfj.8X' JOB

This JCL must be submitted 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
```

List of components with their date of creation

D99INSL MODULE: '\$prfj.D9' 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, 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
/*

```

Installation of the Development Database

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

You do so via the Administrator workbench. (refer to chapter 'Installation of Client Components', subchapter 'Administrator & Developer workbench').

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

Allocation and Loading of Database Parameters

Module I01SY : Job '\$prfj.I1'

Note :

This PDS contains the definitions of the database files. These sizes are intended for the installation tests database. They should be altered by the user in keeping with the evolution of the database size.

Step	Program	Comments
STEP1	IDCAMS	DELETE 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: Job '\$prfj.I2'

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  *
//*  ->NOTE 1                              *
//*  -----                              *
//*  REPLACE :/ BY ./ BEFORE SUBMITTING  *
//*  ->NOTE 2                              *
//*  -----                              *
//*  THIS JOB CONTAINS THE SYSIN'S FOR   *
//*  THAT MAKING UP THE DEVELOPMENT DATA *
//*  ADAPT THE SIZES SPECIFIED IN THE ' *
//*  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 ($VCAT) */
:/      ADD NAME=DF$BASE.WY
DELETE ($INDUV..$BASE.WY) CLUSTER
SET LASTCC = 0
SET MAXCC = 0
DEFINE CLUSTER ( NAME ($INDUV..$BASE.WY) -
        SHR (2 3) RUS NUMBERED -
        VOL ($VOLUV) -
        CYL (4 1) -
        RECSZ (310,310) ) -
DATA    ( NAME ($INDUV..$BASE.WY.D) -
        FSPC (10,5) -
        CISZ (4096) ) /*: CATALOG ($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) */

```



```

        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)
:/ ADD NAME=DF$BASE.WY
DELETE ($INDUV..$BASE.WY) CLUSTER
SET LASTCC = 0
SET MAXCC = 0
DEFINE CLUSTER ( NAME ($INDUV..$BASE.WY) -
                SHR (2) RUS NUMBERED -
                VOL ($VOLUV) -
                CYL (4 1) -
                RECSZ (310,310) ) -
DATA ( NAME ($INDUV..$BASE.WY.D) -
      CISZ (4096) ) /*: CATALOG ($VCAT) */
:/ ADD NAME=DL$BASE.MY
DELETE ($INDUV..$BASE.MY) NONVSAM
PP
//

```

Initialization of Generation Data Groups

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

Module I03DEF : '\$prfj.I3' JOB

Definition of the database files

Step	Program	Comments
STEP1	IDCAMS	Delete/Define AN
DELAR	IDCAMS	Delete AR
DEFAR	IEBFR14	Define AR
DELAJ	IDCAMS	Delete AJ
DEFAJ	IEBFR14	Define AJ
DELAY	IDCAMS	Delete AY

Step	Program	Comments
DEFAY	IEBFR14	Define AY
INITDC	LDDC	Initializes DC
INITPA	LDPA	Initializes PA

EXECUTION JCL

```

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

```

I03INI MODULE: '\$prfj.I3I' JOB

Allocation of generation files (GDG)

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,(220,10),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 3.0, 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
Y04000 EN
//
```

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 FH 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 chapter 'Components', subchapter 'Development Database', Section 'Pac/Impact'.

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='$VCAT',                USER VSAM CATALOG
//*:      SYSCAT='$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=$UWK                       WORK UNIT
//*****
//INIF01 EXEC PGM=BVPRMSYS,PARM='&INDUN..&USER..&BASE'
//*-----
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//        DD DSN=$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)
//INIF02 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)
//INIF03 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
//*-----
/*:STEP CAT 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
//INIF01.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:
`www.microsoft.com/msdownload/vbscript/scripting.asp`
- Both Administrator & Developer workbench and eBusiness Tools components require, for their online help, that Netscape or Internet Explorer 5.5 be installed.
- 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 indicate when you install a Client Component will have to be strictly reused when these Databases are installed at the server level.

Root Directory

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

```
C:\Program Files\IBM\VisualAgePacbase
```

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 installer will propose:

- either a refresh under the initial root directory,
- or a new root, which cannot be modified, built from the previous root. The name of its last directory will be numerically incremented.

For example if you did not modify the default root upon the initial installation, the root of the first re-installation will be:

```
C:\Program Files\IBM\VisualAgePacbase_1
```

And the root of the second re-installation will be:

```
C:\Program Files\IBM\VisualAgePacbase_2
```

Installation Startup

Insert the installation CD.

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 licence. You agree with the terms of the licence ; 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 or the MVS CICS CPI-C, depending on which protocol is used at the server level (see the Installation of Server Environment chapter).

- If it runs on MVS/IMS, you can either use the MVS IMS Connect Socket or the MVS IMS CPI-C, depending on which protocol is used at the server level (see the Installation of Server Environment chapter).

NOTE: With MVS CICS CPI-C or MVS IMS CPIC-C, you must install and configure this protocol on the (or all) workstation(s) where the VisualAge Pacbase Middleware is installed.

For more information on this configuration, see subchapter 'Additional Information', section 'How to configure the MVS CICS/IMS CPI-C protocol'.

THE MIDDLEWARE

The middleware must always be installed on each Developer workstation. 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, eBusiness Tools and Pacbase Web Connection. For the latter, this installation is automatically launched if the Context Server option is selected.

The middleware installed on each Developer workstation then 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.

This option is not available with the Pacbase Web Connection component.

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 the JDK (Java Developer's Kit) 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 select the one or more elements to install:

- Administrator workbench

WARNING: Installing the Administrator workbench on at least one workstation is **REQUIRED** as it will allow for the creation of the site's VA Pac Database(s), Libraries, Profiles, users, etc.

- Developer workbench

Developer workbench includes the following modules:

- Batch module,
- eBusiness module and three eBusiness tools:
 - Proxy Generator,
 - Location Editor,
 - Services Test Facility.

NOTE: These tools are installed following the installation of the rest of the eBusiness module.

- Rational Rose Bridge.

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 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.

NOTE: You will also have to install the middleware and configure the communication on the intermediate server which hosts the gateway.

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.

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

Middleware: 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 box is selected, VA Pac authentication is performed by the security system.

If both boxes are selected, 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_batch.bat
- wb_eBusiness.bat
- wb_cfm.bat

These files are to be found in the root directory of the Administrator and Developer workbench (adworkbench).

The cfm module is a utility that 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 at your site. Consult the VisualAge Support team for more information.

START MENU / PROGRAMS CHOICE

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

Administrator-Developer workbench

- Administration
- Batch
- eBusiness
- cfm

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 from the VA Pac Support web page at:

<http://www.ibm.com/software/ad/vapacbase/support.htm>

In the "Support downloads" box, click the "VisualAge Pacbase downloads" link. In the new page, select the Open Jade and Tidy" download.

You obtain a zipped file called Adwb_util_3.0.exe that you unzip in the installation root directory, by default:

C:\Program Files\IBM\VisualAgePacbase

eBusiness Tools

The eBusiness Tools are:

- Proxy Generator
- Location Editor
- Services Test Facility
- VisualAge Pacbase Connector

This installation allows the eBusiness Tools to be used independently of Developer workbench, without a connection to the VisualAge Pacbase server. eBusiness tools are installed as VisualAge for Java features and tools. The specificity of VisualAge Pacbase Connector is to run in VisualAge for Java only.

If the JDK (Java Developer's Kit) 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 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 eBusiness Tools component can run via a gateway. In this case you will have to install the Middleware component and configure the communication at the level of the intermediate server which hosts the VisualAge Pacbase gateway.

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

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.

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

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

eBusinessTools\bin

NOTE: VisualAge Pacbase Connector runs only as a tool in VisualAge Java.

START MENU / PROGRAMS CHOICE

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

eBusinessTools
 Location Editor
 Proxy Generator
 Services test Facility

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.

NOTE: You will also have to install the middleware and configure the communication on the intermediate server which hosts the gateway.

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.

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

Middleware : 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).

START MENU / PROGRAMS CHOICE

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

```
WorkStation
    WorkStation
    WorkStation 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      [300F]
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
FAA	IFW
ADM	SSADM (in English only)
OMT	OMT

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

Pacbase Web Connection

INSTALLATION PREREQUISITE

For Pacbase Web Connection, you need to install a PERL interpreter (version 5.0 minimum) that you can download from the following URL:

<http://www.perl.com>

To install the interpreter, follow the indications given below.

INSTALLATION

The choice of the installation root for Pacbase Web Connection depends on the installation context. For more details, consult the first page of this chapter.

Then, you install the context server or Pacbase Web Generator or both.

You can install the server context and the generator on two different workstations or on the same developer workstation from where the html pages will be generated.

Furthermore, the PERL interpreter must be installed on the workstation that will be used to launch the HTTP server. The PERL installer will handle the script that sets up the connection between the HTTP server and the context server.

NOTE: The HTTP server and the context server can be installed on different workstations.

The installer asks you to enter the name of the directory where the PERL interpreter is installed.

If PERL is not installed yet, the installer creates the corresponding directory, called Pacwebperl. A BIN sub-directory necessarily contains the cgicgi.pl and cgi-lib.pl files.

NOTE: The actual installation can now start ; press the [Install] button.

The installation of the Context Server is automatically followed by the installation of the Middleware component except if it is already installed under the root of the current installation. When installing the Middleware component, you must specify certain communication parameters.

For information on this part of the installation, see sub-chapter Middleware.

NOTE: IMPORTANT information concerning communication is given at the beginning of the chapter.

EXECUTION FILE

The Pacbase Web Connection execution file is:

Pacweb.exe

This file is located in the PacWeb directory under the root where Pacbase Web Connection is installed.

INSTALLATION OF THE CONTEXT SERVER AS AN NT SERVICE

The context server can be installed under Windows/NT as an NT service. In this case, the context server starts automatically or can be started via the dialog box which is used for all services.

To install the context server as a service, enter the following command from the context server installation directory:

```
pacweb -i [<socket nb>]
```

The socket number is optional. The default value is 2345.

At installation time, the context server is not started up automatically ; it must be done by pressing the start button in the services dialog box.

When the computer boots up, it is automatically executed.

To uninstall the service, enter the following command:

```
pacweb -d [<socket nb>]
```

These commands used to install or uninstall services can be executed only by a user who is authorized to open a session as a service.

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.

NOTE: For the Pacbase Web Connection component, the installation of the Context Server sub-component starts the installation of the Middleware.

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.

- If communication is via the VisualAge Pacbase gateway, these parameters will be requested during the installation of the Middleware on this intermediate server.
- If communication is direct, these parameters will be requested during the automatic installation of the Middleware.

LIST OF PARAMETERS

- VAP SOCKET
IP address: IP address and port used by the VA Pac server
- IMS SOCKET
IP address: IP address and port used by the VA Pac server
Transaction code: Code of the IMS transaction Monitor.
Code Page: Value identifying the coding of characters used by the VA Pac server. 1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MVS CICS/IMS 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.
Protocol code page: Value identifying the coding of characters used by the VA Pac server. 1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MVS IMS Connect
IP address: IP address and port used by the VA Pac server.
Transaction code : IMS transaction code of the VA Pac Communications Monitor.

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

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

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

Additional Information

How to Configure the IMS ITOC CPI-C Protocol

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

The prerequisites are:

- IBM Personal Communications 4.2
- Microsoft SNA Server 3.0A
- Service Pack 2 must be installed for APPC corrections.

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

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

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

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

Example of location in vaplocat.ini:

```
<SocketIMS>
LENGTH=31744
MONITOR=XXXXXX
MWARE=TCPIMS
MWCODEPAGE=297
MWTIMEOUT=30
IXO_ADDRESS=9.144.0.249 4000
IXO_TRANSID=WKV3
IXO_DATASTORE=IMSC
IXO_RACFGROUP=FR4012
```

In the "IXO_TRANSID" field, type the IMS transaction code which is associated with the BVPSCPI communication monitor.

- To define the VA Pac IMS transaction, you need to declare it on IMS via a GEN INPUT IMS as follows:

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

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\VisualAgePacbase\Common\

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.

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 personalized 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.

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 et + : 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.
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 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\VisualAgePacbase\Common\

NOTE: 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.

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 3000
```
- IMS SOCKET

```
<LocationName>
COMM_TYPE=TCPIMS
MONITOR=BVPSSOC
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=297
IXO_ADDRESS=127.0.0.1 3000
IXO_TRANSID=P300CPI
```
- MVS CICS/IMS CPI-C

```
<LocationName>
COMM_TYPE=CPIC
MONITOR=BVPSCPI
MESSAGE_LENGTH=8192
IXO_TIMEOUT=30
HOST_ENCODING=1147
```
- IMS Connect

```
<NomLocalisation>
COMM_TYPE=TCPIMS
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
HOST_ENCODING=1147
IXO_ADDRESS=127.0.0.1 3000
IXO_TRANSID=P300CPI
IXO_DATASTORE=IMSC
IXO_RACFGROUP=FR42
```

DETAILS ON THE PARAMETERS

The following list is organised 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 or MVS/IMS, with the use of a TCP/IP listener.

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

- **IXO_ADDRESS:**
IP address and port used by the VA Pac Server. The port number must correspond to the one indicated at the server configuration.
- **IXO_DATASTORE:**
Name of link to IMS defined in IMS Connect (IMS DataStore ID)
- **IXO_RACFGROUP:**
Name of RACF group for IMS Connect.
- **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:**
CICS transaction code
- **HOST_ENCODING:**
Identifies the encoding of the characters used by the VisualAge Pacbase server.
1140 (US EBCDIC) or 1146 (UK EBCDIC)
- **MESSAGE_LENGTH:**
This parameter's value must be 31744.
- **MONITOR:**
Communication monitor code for VisualAge Pacbase, which is BVPSCPI, or BVPSSOC for MVS CICS SOCKET.

NOTE: For MVS CICS/IMS 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.

Uninstalling Client Components

To uninstall a Client component, use the Windows service "Add/Remove" Programs in the Configuration Panel.

You can also use the Installation CD and activate the "Remove" function.

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
JCLARCH	Storage of the Development Database journal
JCLSAVE	Saving of the Development Database
JCLMLIB	Library management
JCLREOR	Reorganisation 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,
- Extraction utility tests.

Generation-Print, TP 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.
- 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.

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.

- 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.

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

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

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

Each of these jobs can be followed by a UPDT 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 Pac" 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: the installation JCLs have been kept,

Case 2: the installation JCLs must be regenerated for a standard reinstallation,

Case 3: the installation JCLs must be regenerated

for a non-standard reinstallation.

CASE 1: INSTALLATION JCLs HAVE BEEN KEPT

The standard re-installation procedure consists of executing the jobs contained in the following JCL modules:

- 1) D04MBR : \$prfj.D4B job: renaming exit-users batch load-
1. D04PSREN : job \$prfj.D4P renaming PSB TP,
2. D04SRCD: \$prfj.D4SR job: updating DBD sources and possibly recompiling DBDs,
3. D04SRCP: job \$prfj.D4SR: updating PSBs sources and possibly recompiling PSBs,

- 2) D05PROC : job \$prfj.D5 loading procedures,
- 2) D06SKEL: \$prfj.D6 job: loading skeleton files,
- 3) D07AE0 : \$prfj.D7 job: loading error messages,
- 4) D08XMET: \$prfj.D8X job: installation of the administration model, (see the description in the appendix at the end of the manual).
- 5) I05META: \$prfj.I5 job: 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).

NOTE& If the \$prfj.D8X job report indicates that a re-organization is requested, you will have to run the following jobs:

- 1) JCLSVAD: \$prfj.SAD job: backup of the administration Database,
- 2) JCLROAD: \$prfj.OAD job: re-organization of the administration Database,
- 3) JCLARAD: \$prfj.AAD job: archiving of the administration Database journal,
- 4) JCLRSAD: \$prfj.RAD job: restoration of the administration Database.

If the \$prfj.I5 job report indicates that a re-organization is requested, you will have to run the following jobs:

- 1) JCLSAVE: \$prfj.SAV job: backup of the development Database,
- 2) JCLREOR: \$prfj.REO job: re-organization of the development Database,
- 3) JCLARCH: \$prfj.ARC job: archiving the development Database journal,
- 4) JCLREST: \$prfj.RES job: restoration of the development Database.

CASE 2: INSTALLATION JCLS MUST BE REGENERATED FOR A STANDARD REINSTALLATION

For detailed information, refer to chapter 'Server Installation', subchapter 'Server Installation Parameters'.

The re-generation of JCLs consists of executing the BVPMMJCL utility again with the parameters selected for the installation on-site, and adding the JCLs necessary to the re-installation process.

Check the resulting JCLs.

Proceed with the installation according to the steps described in CASE 1.

CASE 3: INSTALLATION JCLS MUST BE REGENERATED FOR A NON-STANDARD REINSTALLATION

For detailed information, refer to chapter 'Server Installation', subchapter 'Server Installation Parameters'.

The re-generation of JCLs consists in executing the BVPMMJCL utility again with the parameters selected for the installation on-site, and adding the JCL's necessary to the re-installation process. These JCL's are indicated in the note enclosed with the version tape.

Once the JCLs are completed, follow the instructions of case 1 (for a standard re-installation) and the note enclosed with the version cartridge (or tape).

Chapter 7. Retrieval

Retrieval of VisualAge Pacbase 2.0 and 2.5

Operations to be Performed

The installation of the 3.0 release 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 six 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 3.0 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 (enter the access key),
 - D08XMET: installation of the Administration Model (see the Appendix, at the end of the manual),
- re-organization of the Administration Database if it is mentioned in the execution report of the preceding job (prfjD8X).
 - 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,
 - 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,
- JCLREOR: re-organization of new Development Database,
- I04REST: restoration of the new Development Database from the backup obtained previously,
- I05META: installation of the new Database development Model (see the description at the end of the manual).

The execution of the following JCL's is optional and it is sometimes required for a better optimisation.

- 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 and 6 are optional.

4) Retrieval of generation-print commands.

It involves executing the following JCL:

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

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 in which you indicate the Administrator's code and name.

If the Administrator's code or name is not indicated, an error message is sent and the procedure cannot be executed.

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 in which you indicate the characteristics of the development Databases which are to 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

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	&&PACGY	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		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

Code	Physical name	Type	Label
B7DC\$BASE B7D3\$BASE	&DSMS DUMMY DUMMY	Input Input	DSMS file of Development Database elements 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=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&PE25MB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGY DD DSN=&INDSV..BVPGY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//*-----
//PTU920 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU920,PTU920$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT 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=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPAF900,GAF900$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
//*:STEP CAT 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..BVP7AE,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,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=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPACA15,ZACA15$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

PRINCIPLE

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

EXECUTION CONDITIONS

None

PRINTED OUTPUT

This procedure prints a report which indicates the number of Manuals changed into Volumes, the warnings on User Entities, calls of Parameterized Input Aids and description of Reports (long data), the code of the new development Database and the number of records output by the PC file.

RESULT

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

PC25 - Notes on Data Retrieval

SPLITTING UP OF THE COMMENT DESCRIPTION (-G)

The comment description is split up into several descriptions.

- Comments

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

CAUTION

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

- Generation lines

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

- Generation parameters

They include the O line type (-GO).

- Error messages management

They include the C, D, F ,S ,T , U line types (-GE).

- Call of entities via Relations

They include the R line type (-CR).

- Specificity of the Input Aid entity

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

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.

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 'EIBM'.

LONG DATA: USER ENTITIES, INPUT AIDS, REPORT LAYOUTS

There are no more continuation records for these entities. Formally, there was one index for one main record and one index for each continuation record. Now long data is created to concatenate the information included in the previous records. This data can be 1,000 characters long. It is split up into several records. Now a single index is created and it points at the first of these records.

IMPORTANT: If a continuation record is modified in a sub-library, in the 2.n release, this modification is not retrieved in the long data in the lower libraries.

So if the retrieval process detects any update in one or more sub-library(ies) on the same continuation record, the following warning messages are displayed:

- "Risk of inconsistency of the \$xx xxxxxx user entity definition in the xxx library".
- 'Risk of inconsistency of the xxx documentation line under xxxxxxxx input aid in the xxx library'.
- "Risk of inconsistency of labels on the xxx line of the xxx report in the xxx library"

The user will have to modify these records manually if they are inconsistent with those of the 2.0 or 2.5 release.

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	Firtst data record (length=153)
PAC7PE	&&PE	Output	User Entity Occurence 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 Occurences (2.5 release)
PAC7PF	&&PF	Output	Other records (length=153)
PAC7PM	&&PM	Output	Report file (length=48)
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
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

COMMENTS PROCESSING: PTU92A

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
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	

META-ENTITIES PROCESSING: PTU912

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVPAE	Input	Error messages
PAC7PZ	&&PZ	Input	User Entities (2.5 release)
PAC7PB	&&PB	Input	First data record
PAC7ZP	&&ZP	Output	Development Model records (Definition and Descriptions) (length=193)

Code	Physical name	Type	Label
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 Occurences Definition (2.5 release), Report layouts and Comments (except calls of input aids)
PAC7PB	&&PB	Input	First data record
PAC7PH	&&PH	Input	Description of Input Aids and of their calls in the Comments
PAC7PM	&&PM	Input/Output	Report file
PAC7ZP	&&EP	Output	User entity Occurences Definition (2.5 release), Report layouts, and Comments (call of Input Aids included) (length=193)
PAC7PO	&&PO	Output	Comments (except the call of Input Aids) (length=153)
PAC7PD	&&PD	Output	First data record (length=153)
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 Occurences (2.5 release), Report layouts, and Comments (including the call of Input Aids)

Code	Physical name	Type	Label
PAC7PZ	&&ZP	Input	Definition and Description of the Development Model and Description of User Entity Occurrences (2.5 release)
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	&&PD	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 Input	Intermediate long data
SORTOUT	&&XZ	Output Output	Sorted long data (length=193)
SORTWK01		Sort	
SORTWK02		Sort	
SORTWK03		Sort	

FILES MERGING: PTU914

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

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP7AE	Input	Error messages
PAC7ZP	&&XZ	Input	Sorted long data
PAC7PO	&&PO	Input	Comments (no call of Input Aids)
PAC7PD	&&PD	Input	First data record

Code	Physical name	Type	Label
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 (3.0 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='$PRCLIB',                IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$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=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU911,PTUREP$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//PAC7MB DD DSN=&&PC25MB,DISP=(OLD,DELETE)
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MC DD DSN=&OLDPC,DISP=SHR
//PAC7PF DD DSN=&&PF,DISP=(NEW,PASS),UNIT=&UWK,
//      SPACE=(CYL,&CYL,RLSE),
//      DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PB DD DSN=&&PC,DISP=(NEW,PASS),UNIT=&UWK,
//      SPACE=(TRK,1,RLSE),
//      DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PE DD DSN=&&PE,DISP=(NEW,PASS),UNIT=&UWK,
//      SPACE=(CYL,&CYL,RLSE),
//      DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PG DD DSN=&&PG,DISP=(NEW,PASS),UNIT=&UWK,
//      SPACE=(CYL,&CYL,RLSE),
//      DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PL DD DSN=&&PL,DISP=(NEW,PASS),UNIT=&UWK,
//      SPACE=(CYL,&CYL,RLSE),
//      DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PM DD DSN=&&PM,DISP=(NEW,PASS),UNIT=&UWK,
//      SPACE=(CYL,&CYL,RLSE),
//      DCB=(RECFM=FB,LRECL=48,BLKSIZE=4800)
//PAC7PZ DD DSN=&&PZ,DISP=(NEW,PASS),UNIT=&UWK,
//      SPACE=(CYL,&CYL,RLSE),
//      DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7ET DD SYSOUT=&OUT
//*-----
//PTU909 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU909,PTUREP$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7PB   DD DSN=&&PC,DISP=(OLD,PASS)
//PAC7PL   DD DSN=&&PL,DISP=(OLD,DELETE)
//PAC7PI   DD DSN=&&PI,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PM   DD DSN=&&PM,DISP=(OLD,PASS)
//*-----
//PTU92A   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU92A,PTUREP$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=((0,NE,PTU911),(0,NE,PTU909))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT 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..BVP7AE,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)
//*-----
//PTU912   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU912,PTUREP$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=((0,NE,PTU911),(0,NE,PTU909),(0,NE,PTU92A))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE  DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7PZ   DD DSN=&&PZ,DISP=(OLD,DELETE)
//PAC7ZP   DD DSN=&&ZP,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//PAC7PB   DD DSN=&&PC,DISP=(OLD,PASS)
//*-----
//PTU919   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU919,PTUREP$SUG,&BUF,

```

```

//      &SPIE&TEST&EXCPVR&RST,&PRLD,
//      &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//      &IRLM),
//      COND=(0,NE,PTU911),(0,NE,PTU909),(0,NE,PTU92A),
//      (0,NE,PTU912))
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//SORTLIB  DD DSN=&SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,&CYL,,CONTIG)
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7PB   DD DSN=&&PC,DISP=(OLD,DELETE)
//PAC7PE   DD DSN=&&PE,DISP=(OLD,DELETE)
//PAC7PH   DD DSN=&&PH,DISP=(OLD,DELETE)
//PAC7PM   DD DSN=&&PM,DISP=(OLD,PASS)
//PAC7PD   DD DSN=&&PD,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7PO   DD DSN=&&PO,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=153,BLKSIZE=27846)
//PAC7ZP   DD DSN=&&EP,DISP=(NEW,PASS),UNIT=&UWK,
//          SPACE=(CYL,&CYL,RLSE),
//          DCB=(RECFM=FB,LRECL=193,BLKSIZE=27985)
//*-----
//PTU913   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU913,PTUREP$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=(0,NE,PTU911),(0,NE,PTU909),(0,NE,PTU92A),
//          (0,NE,PTU912),(0,NE,PTU919))
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$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)
//PAC7PD   DD DSN=&&PB,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),(0,NE,PTU909),
//          (0,NE,PTU92A),(0,NE,PTU912),(0,NE,PTU919),
//          (0,NE,PTU913))
//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=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU914,PTUREP$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=((0,NE,PTU911),(0,NE,PTU909),(0,NE,PTU92A),
//          (0,NE,PTU912),(0,NE,PTU919),(0,NE,PTU913),(0,NE,SORTZX))
//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=&&PB,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 3.0 release 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 2.0 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	&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)
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		Output	Work file (length=372)

Code	Physical name	Type	Label
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..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database Index file
BVP7GR	&INDSV..BVP GR	Input	Administration Database Data file
BVP7GY	&INDSV..BVP GY	Input	Administration Database Extension
BVP7GY	&INDSV..BVP GU	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)
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	&&PACGY	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		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

Code	Physical name	Type	Label
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension
BVP7GU	&INDSV..BVPGU	Input	Administration Database users
B7DC\$BASE B7D3\$BASE	&DSMS DUMMY DUMMY	Input Input	DSMS file of Development Database elements 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

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
//*:       SYSCAT='$SCAT',                  VA PAC SYSTEM VSAM CATALOG
//          STEPLIB='$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='$PRCLIB',                       IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$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
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGY DD DSN=&INDSV..BVPGY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//*-----
//PTU908 EXEC PGM=BVPTU908
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//*:STEP CAT 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=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU921,PTU921$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMT0,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR

```

```

//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7MB   DD DSN=&&PG20MB,DISP=(OLD,DELETE)
//PAC7PG   DD DSN=&&PG,DISP=(OLD,DELETE)
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(100,20),RLSE),
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7GZ   DD DSN=&&PACGZ,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(100,20),RLSE),
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//*-----
//PAF90Y   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPAF900,PAF900$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM),
//          COND=(0,NE,PTU921)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$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=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPACA15,PACA15$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//B7AJ$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=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPAF900,GAF900$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
//*:STEP CAT 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..BVP7AE,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7GY DD DSN=&&PAC7GY,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=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPACA15,ZACA15$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GJ DD DSN=&INDSV..BVP8GJ,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GY DD DSN=&INDSV..BVP8GY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//BVP7GY DD DSN=&INDSV..BVP7GY,DISP=SHR
//B7DC$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 3.0 release 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.

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

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)
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

Code	Physical name	Type	Label
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		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	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
PAC71F		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	&&PACGY	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		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

Code	Physical name	Type	Label
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 B7D3\$BASE	&DSMS DUMMY DUMMY	Input Input	DSMS file of Development Database elements 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

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=$BCOB,DISP=SHR
//CARTE DD DDNAME=SYSIN
//PAC7MB DD DSN=&&PG25MB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//*:          DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVAE,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGY DD DSN=&INDSV..BVPGY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//*-----
//PTU921 EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU921,PTU921$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PAC7MB   DD DSN=&&PG25MB,DISP=(OLD,DELETE)
//PAC7PG   DD DSN=&OLDPG,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(100,20),RLSE),
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7GZ   DD DSN=&&PACGZ,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(100,20),RLSE),
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//*-----
//PAF90Y   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPAF900,PAF900$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,&DBRC,
//          &IRLM),
//          COND=(0,NE,PTU921)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$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..BVP7AE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,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=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPACA15,PACA15$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDR DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AJ$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=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPAF900,GAF900$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
//*:STEPCLIB 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..BVP7AE,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7GY DD DSN=&&PAC7GY,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=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPACA15,GACA15$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//BVP8GJ DD DSN=&INDSV..BVP8GJ,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GY DD DSN=&INDSV..BVP8GY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//BVP7GY DD DSN=&INDSV..BVP7GY,DISP=SHR
//B7DC$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
//*

```

PEI Retrieval (PP25)

PP25 - Introduction

PRINCIPLE

This procedure retrieves the 2.0 or 2.5 PP file, which is the sequential image of the Production Environment Interface, and updates the 3.0 development database.

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 3.0 format, and updates the 3.0 development database.

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.

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
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		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..BVP AE	Input	Error messages
BVP7GN	&INDSV..BVP GN	Input	Administration Database Index file
BVP7GR	&INDSV..BVP GR	Input	Administration Database Data file
BVP7GY	&INDSV..BVP GY	Input	Administration Database Extension
BVP7GY	&INDSV..BVP GU	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)
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,FMT0=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&PP25MB,DISP=(,PASS),
//          UNIT=UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE  DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7AJ  DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN  DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR  DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGY  DD DSN=&INDSV..BVPGY,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//*-----
//PTU923  EXEC PGM=DFSRRC00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU923,PTU923$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:      DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP AE,DISP=SHR
//PAC7MB   DD DSN=&&PP25MB,DISP=(OLD,DELETE)
//PAC7PP   DD DSN=&OLDPP,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(100,20),RLSE),
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//*-----
//PAF900   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPAF900,PAF900$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM),
//          COND=(0,NE,PTU923)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//      DD DSN=&STEPLIB,DISP=SHR
//      DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT 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..BVP7AE,DISP=SHR
//B7AN$BASE DD DSN=&INDUV..&BASE.AN,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,DISP=SHR
//PAC7GY DD DSN=&&PACGY,DISP=(OLD,DELETE)
//PAC7ME DD DSN=&&PAC7ME,DISP=(,PASS),UNIT=&UWK,
//
// DCB=(RECFM=FB,LRECL=372,BLKSIZE=27156)
//PAC7MV DD DSN=&&PAC7MV,DISP=(,PASS),UNIT=&UWK,
//
// DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MW DD DISP=(,DELETE),UNIT=&UWK,
//
// DCB=(RECFM=FB,LRECL=170,BLKSIZE=6120)
//PAC7MX DD DISP=(,DELETE),UNIT=&UWK,
//
// DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//PAC7MY DD DISP=(,DELETE),UNIT=&UWK,
//
// DCB=(RECFM=FB,LRECL=748,BLKSIZE=26928)
//*-----
//PACA15 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//
// PARM=(DLI,BVPACA15,PACA15$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*:
// DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDR DD DUMMY,
//
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP7AE,DISP=SHR
//B7AJ$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 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	&&PG25MB	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
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	&&PACGY	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		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 B7D3\$BASE	&DSMS DUMMY DUMMY	Input Input	DSMS file of Development Database elements 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

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,FMT0=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//INPUT EXEC PGM=BVPTU001
//STEPLIB DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//CARTE   DD DDNAME=SYSIN
//PAC7MB  DD DSN=&&UV25MB,DISP=(,PASS),
//          UNIT=&UWK,SPACE=(TRK,(5,1),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE  DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR  DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN  DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU  DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGY  DD DSN=&INDSV..BVPGY,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
//-----
//PTU922 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU922,PTUREP$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMT0,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR

```

```

//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS      DD DSN=&PSBLIB,DISP=SHR
//          DD DSN=&DBDLIB,DISP=SHR
//*:STEPCAT DD DSN=&SYSTCAT,DISP=SHR
//*:       DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT   DD SYSOUT=&OUT
//SYSOUX   DD SYSOUT=&OUT
//DDSNAP   DD SYSOUT=&OUT
//PROCLIB  DD DSN=&PROCLIB,DISP=SHR
//IEFRDER  DD DUMMY,
//          DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//          BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE   DD DSN=&INDSV..BVP7AE,DISP=SHR
//PAC7MB   DD DSN=&&UV25MB,DISP=(OLD,DELETE)
//PAC7UV   DD DSN=&OLDUV,DISP=SHR
//PAC7GY   DD DSN=&&PACGY,DISP=(,PASS),UNIT=&UWK,
//          SPACE=(TRK,(100,20),RLSE),
//          DCB=(RECFM=FB,LRECL=310,BLKSIZE=3100)
//PAC7ET   DD SYSOUT=&OUT
//*-----
//PAF900   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPAF900,GAF900$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM),
//          COND=(0,NE,PTU922)
//STEPLIB  DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$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..BVP7AE,DISP=SHR
//BVP8GN   DD DSN=&INDSV..BVP8GN,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=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPACA15,GACA15$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
//*:STEP CAT 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..BVP7AE,DISP=SHR
//BVP8GJ DD DSN=&INDSV..BVP8GJ,DISP=SHR
//BVP8GN DD DSN=&INDSV..BVP8GN,DISP=SHR
//BVP8GR DD DSN=&INDSV..BVP8GR,DISP=SHR
//BVP8GY DD DSN=&INDSV..BVP8GY,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,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 MB Transactions (MB25)

MB25 - Introduction

PRINCIPLE

This procedure retrieves the 2.0 or 2.5 MB transactions in the 3.0 release.

REQUISITES

None.

PRINTED OUTPUT

This procedure prints a report of the errors encountered.

RESULT

This procedure generates a transaction file for the 3.0 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

Code	Physical name	Type	Label
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=$IRLM
//*****
//VERIFY EXEC PGM=IDCAM5
//-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*:       DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//-----
//PTU926   EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU926,PTU926$SUG,&BUF,
//          &SPIE&TEST&EXCPVR&RST,&PRLD,
//          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
//          &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
//          DD DSN=&STEPLIB,DISP=SHR
//          DD DSN=$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

```

Retrieval of GY Transactions (GY25)

GY25 - Introduction

PRINCIPLE

This procedure retrieves the 2.0 or 2.5 GY transactions and formats them in the 3.0 release format.

REQUISITES

None.

PRINTED OUTPUT

This procedure prints a report of the errors encountered.

RESULT

This procedure generates a transaction file for the 3.0 UPDP procedure 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	&&PACMV	Output	Retrieval transactions for UPDP
PAC7ME	&&PACME	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

GY25 - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*              RETRIEVAL OF GY FILE
/*
/* -----
/*
//BVPGY25  PROC BASE=$BASE,                CODE OF VAPAC DATABASE
//          OUT=$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='$PRCLIB',             IMS PROCLIB
//          BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
//          CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$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..BVP AE,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PTU927 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU927,PTU927$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
// DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
// BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//B7AR$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

```

Retrieval of MB Transactions (MB30)

MB30 - Introduction

PRINCIPLE

This procedure retrieves the 3.0 MB transactions and formats them in the 2.0 or 2.5 format.

REQUISITES

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	3.0 MB transactions
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='$PRCLIB', IMS PROCLIB
// BUF=40,SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,SRCH=0,
// CKPTID=,MON=N,LOGA=0,FMT0=T,DBRC=$DBRC,IRLM=$IRLM
//*****
//VERIFY EXEC PGM=IDCAMS
//*-----
//*:STEPCAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
//*-----
//PTU928 EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPTU928,PTU928$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMT0,,,&DBRC,
// &IRLM)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$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

```

Retrieval of GY Transactions (GY30)

GY30 - Introduction

PRINCIPLE

This procedure retrieves the 3.0 GY transactions and formats them in the 2.0 or 2.5 format.

REQUISITES

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

Code	Physical name	Type	Label
BVP7GN	&INDSV..BVPGN	Input	Administration database index
BVP7GU	&INDSV..BVPGU	Input	Administration database users
PAC7GY	&OLDGY	Input	3.0 GY transactions
PAC7MV	&&PACMV	Output	Retrieval transactions for UPDP
PAC7ME	&&PACME	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

GY30 - Execution JCL

```

/** -----
/**      VISUALAGE PACBASE
/**
/** -----
/**      RETRIEVAL OF GY FILE
/**
/** -----
/**
//BVPGY30  PROC BASE=$BASE,                CODE OF VAPAC DATABASE
//          OUT=$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
//*:       SYSCAT='$SCAT',                VA PAC SYSTEM VSAM CATALOG
//          STEPLIB='$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=&SYSCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PAC7AE   DD DSN=&INDSV..BVPAE,DISP=SHR
//PACGGR   DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGN   DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGU   DD DSN=&INDSV..BVPGU,DISP=SHR
//SYSIN    DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
//          DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
/**-----

```

```

//PTU929 EXEC PGM=DFSRR00,REGION=$REGSIZ,
//      PARM=(DLI,BVPTU929,PTU929$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=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
//      DD DSN=&DBDLIB,DISP=SHR
//*:STEP CAT DD DSN=&SYSTCAT,DISP=SHR
//*: DD DSN=&VSAMCAT,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//SYSOUX DD SYSOUT=&OUT
//DDSNAP DD SYSOUT=&OUT
//PROCLIB DD DSN=&PROCLIB,DISP=SHR
//IEFRDER DD DUMMY,
//      DCB=(RECFM=VB,BLKSIZE=1920,LRECL=1916,BUFNO=2)
//SYSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSUDUMP DD SYSOUT=&OUT,DCB=(RECFM=FBA,LRECL=121,
//      BLKSIZE=605),SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON DD DUMMY
//DFSVSAMP DD DSN=&INDSN..BVPSY(DFSVSAM8),DISP=SHR
//BVP7AE DD DSN=&INDSV..BVP AE,DISP=SHR
//B7AR$BASE DD DSN=&INDUV..&BASE.AR,DISP=SHR
//BVP7GN DD DSN=&INDSV..BVP GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP GU,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

```

Retrieval of journal file PJ (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 a 3.0 format.

PREREQUISITES

None.

PRINTED OUTPUT

This procedure generates a transaction file indicating the number of transactions retrieved in their initial format 2.n, and the number of transactions converted in a 3.0 format.

RESULT

This procedure generates a PJ journal file in a 3.0 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 a 3.0 version.

Sometimes, to retrieve some transactions, it is necessary to search for information in a 3.0 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
			Journal file old version
PAC7PJ		Input	
			Error messages
PAC7AE		Input	
			Development Database data
PAC7AR		Input	
			Development Database Index
PAC7AN		Input	
			Development Database extension data
PAC7AY		Input	
			Journal file in a 3.0 format (length=170)
PAC7JP		Output	
			Report

Code	Physical name	Type	Label
PAC7ET		Report	

PJ25 - Execution JCL

```

/* -----
/*      VISUALAGE PACBASE
/*
/* -----
/*              RETRIEVAL OF PJ FILE
/*
/* -----
/*
//BVPPJ25  PROC BASE=$BASE,                CODE OF DEVPT DATABASE
//          OUT=$OUT,                       OUTPUT CLASS
//          INDUV='$INDUV',                 INDEX OF USER VSAM FILES
//          INDUN='$INDUN',                 INDEX OF USER NON 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..BVAE,DISP=SHR
//PAC7AN   DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR   DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY   DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN   DD DSN=&INDSN..BVPSY(VERIFAE),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
//         DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//*-----
//PTU918  EXEC PGM=DFSRR00,REGION=$REGSIZ,
//          PARM=(DLI,BVPTU918,PTU918$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
//

```

Procedures - Summary Table of Changes

LIST OF NEW PROCEDURES

Procedure	Comments
ARAD	Archiving Administration Database
ARPM	Archivage du journal du Pont générique
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
INQJ	Initialization of the VA Pac interface archive journal and QJ configuration management
ROAD	Re-organization of Administration Database
RSAD	Restoration of Administration Database
PACG	Backup of Administration Database
VING	Update of Administration Model
PACS	Management of development Database
GY25	Retrieval of GY file for UPDP
MB25	Retrieval of MB file for UPDT
GY30	Retrieval of GY file for UPDP PDP to 2.5

Procedure	Comments
MB30	Retrieval of MB file for UPDT PDT to 2.5
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 2.5
UV25	Retrieval of PacTransfer parameters (UV)
PP25	Retrieval of PEI environment (PP)
ARPM	Archivage du journal Interface Va Pac - Gestion de configuration
CHPM	Environment and Database elements check report
CPPM	Comparison between Database and user configuration
EXPM	Extraction of environments
HIPM	Implementation of elements
SIPM	Generation simulation
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	
MLIB	PTU100 PTU120	Integrated in PACS procedure
SASN	PTU130 PTU140	Integrated in PACS procedure
SAVE	PTU500	Integrated in PACS procedure
UXSR	UTIXSR	Integrated in PACS procedure
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACT11	
GET1	PACT41	
GET2	PACT41 PACT51	

Procedure	Program	Comments
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	
RVKE	PREI40 PREI50	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	
VPU2	PVA320	
LVBL	PTULVB	
QREO	PTUN00 PTUN10 PTUN40	
RPPG	PTU908	
RPTD	PTAR20	

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 DBDs,
- One library for 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 DSNAMES allocation is performed according to the following principles:

- the first index level(s) is/are represented by: the symbolic parameter &INDSV or &INDUV if the organization of the file is VSAM, by the symbolic parameter &INDSN or &INDUN for non-VSAM files, or by the symbolic parameter &INDUN for generation files. 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 always of 6 characters; it consists of two types of files:
 - BVP.xx 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.

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
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 'GPRT' (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

Characteristics	Value
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..BVPDV
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 (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)

Characteristics	Value
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
Recreate	4,605
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
Recreate	4605
CI Size	5120 (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
Recreate	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
Recl	4,605
CI Size	5,120 (data) 1,536 (index)
Key	5 (position 0)
Utilization	Batch only
Dsname	&INDSV..BVPSS

Parameters

Bibliothèque des paramètres systèmes

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	Special indication
DFBVPAE	DELETE/DEFINE AE	
DFBVPGJ	DELETE/DEFINE GJ	
DFBVPGN	DELETE/DEFINE GN	
DFBVPGR	DELETE/DEFINE GR	
DFBVPGS	DELETE/DEFINE GS	
DFBVPGU	DELETE/DEFINE GU	
DFBVPGY	DELETE/DEFINE GY	
DFBVPLB	DELETE/DEFINE LB	
DFBVPPA	DELETE/DEFINE PA	Updatable (1)
DFBVQJ	DELETE/DEFINE QJ	
DFBVPSK	DELETE/DEFINE SK	
DFBVPSL	DELETE/DEFINE SL	
DFBVPSM	DELETE/DEFINE SM	
DFBVPSN	DELETE/DEFINE SN	
DFBVPSO	DELETE/DEFINE SO	
DFBVPSQ	DELETE/DEFINE SQ	
DFBVPSR	DELETE/DEFINE SR	
DFBVPSS	DELETE/DEFINE SS	
DFBVPTR	DELETE/DEFINE TR	
DFBVPSW	DELETE/DEFINE SW	
DFBMSDC	DELETE/DEFINE DC	DSMS: Development Database element (2)
DFSYIANA	DEFINE	Updatable
DLSYIANA	DELETE	
DFSYSEXT	DEFINE	Updatable (3)
DLSYSEXT	DELETE	
DFSYSPAF	DEFINE	Updatable (4)
DLSYSPAF	DELETE	
DFSYTRDU	DEFINE	
DLSYTRDU	DELETE	
DFSYTRPF	DEFINE	
DFWKREOR	DEFINE	
DLWKREOR	DELETE	

Member	Contents or format	Special indication
DFWYREOR	DEFINE	
DLWYREOR	DELETE	
DFWKROAD	DEFINE	
DLWKROAD	DELETE	
DFWYROAD	DEFINE	
DLWYROAD	DELETE	
DFTABTDF	DELETE/DEFINE TD	TABLES: Description
DLPQCE	DELETE	
KEY01	00000001	
LIBVPAE	LISTCAT of AE	
LIBVPGJ	LISTCAT of GJ	
REWYMY	REPRO	
VERIFAE	VERIFY (PAC7AE)	
VERIFAJ	VERIFY (PAC7AJ)	
VERIFAN	VERIFY (PAC7AN)	
VERIFAR	VERIFY (PAC7AR)	
VERIFAY	VERIFY (PAC7AY)	
VERIFEM	VERIFY (PAC7EM)	
VERIFGJ	VERIFY (PAC7GJ)	
VERIFGN	VERIFY (PAC7GN)	
VERIFGR	VERIFY (PAC7GR)	
VERIFGU	VERIFY (PAC7GU)	
VERIFGY	VERIFY (PAC7GY)	
VERIFLB	VERIFY (PAC7LB)	
VERIFQJ	VERIFY (PAC7QJ)	
VERIFSC	VERIFY (PAC7SC)	
VERIFSG	VERIFY (PAC7SG)	
VERIFSN	VERIFY (PAC7SN)	
VERIFSP	VERIFY (PAC7SP)	
VERIFSR	VERIFY (PAC7SR)	
VERIFSS	VERIFY (PAC7SS)	
VERIFTD	VERIFY (PAC7TD)	
SRTPC25	SORT FIELDS	

Member	Contents or format	Special indication
SRTREO1	SORT FIELDS	
SRTREO2	SORT FIELDS	
MAXKEY	maxi. records	
REPRO	IDCAMS input	
REPRO999	IDCAMS input	

- (1) The size depends on the number of PAF requests sent when using the file in the on-line mode.
- (2) must be used if DSMS is to be installed. installed.
- (3) Size depending on the number of requests sent by the deep extractor.
- (4) Size depending on the PAF requests sent in the batch mode when using the file.

Batch Procedure JCL Libraries

The procedures must be loaded in a special library called PROCLIB.

This library can be an existing library or one specially created for the purpose.

In some cases, its characteristics must be the following ones:

Characteristics	Value
Size	Approx.150 blocks of 6,080 bytes
Organization	PDS
DCB	Recfm=FB, Lrecl=80, Blksize=6,080
Dsname	user defined.

Administration Database

Administration Database Files

- Data file, GR.

Characteristics	Value
Organization	VSAM-RRDS
Reclsize	144
CI size	4,096
Utilization	Batch and on-line
Dsname	&INDSV..BVPGR
Size	27 records per C.I. of 4,096

- Extension data file, GY.

Characteristics	Value
Organization	VSAM-RRDS
Reclsize	1018
CI size	4,096
Utilization	Batch and on-line
Dsname	&INDSV..BVPGY
Size	4 records per C.I. of 4,096

- Index file, GN.

Characteristics	Value
Organization	VSAM-KSDS
Reclsize	59
CI size	4,096 (for index and data)
Key	49 (position 0)
Utilization	Batch and TP
Dsname	&INDSV..BVPGN
Size	62 records per C.I. of 4,096

- Journal file, GJ.

Characteristics	Value
Organization	VSAM-RRDS
Reclsize	170
CI size	4096
Utilization	Batch and TP
Dsname	&INDSV..BVPGJ
Size	23 records per C.I. of 4,096

- User file, GU.

Characteristics	Value
Organization	VSAM-RRDS
Reclsize	80
CI size	4,096
Utilization	Batch and on-line

Characteristics	Value
Dsname	&INDSV..BVPGU
Size	50 records per C.I. of 4,096

- Journal file of Standard Bridge QJ

Characteristics	Value
Organization	VSAM-RRDS
Reclsize	1,105
CI size	1,105
Utilization	Batch and on-line
Dsname	&INDSV..BVPQJ
Size	4 records per C.I. of 4,096

Administration Database Backup

The administration Database backup consists of two sequential generation files.

- Backup of the Database (PE).

This is a backup file of the administration Database components: index (GN), Data (GR) and extension (GY) in a sequential format.

Characteristics	Value
Organization	Sequential, generation
Dsname	&INDSN..BVPPE(n)
DCB	Recfm=VB, Lrecl=1,023, Blksize=27,998
Utilization	batch
Size	144 bytes per data
	1,023 bytes per extension data
	59 bytes per index

- Backup of the journal (PK).

The purpose of this file is to store all 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	&INDUN..BVPPK(n)
DCB	Recfm=FB, Lrecl=170, Blksize=6800
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 (TP)
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
DBDName	BDAN\$BASE (batch) TDAN\$BASE (TP)
Lng SEG	59

Characteristics	Value
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 (TP)
Lng SEG	178
Utilization	Batch and on-line
Size	2 records per C.I. of 512

Development Database Parameter 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=1023, Blksize=27998
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=1023, Blksize=27998
Utilization	Batch
Size	59 bytes per index

- The backup of the Development Database extension data (PY) if 'Dispatch' option.

Characteristics	Value
Organization	Sequential, generation
Dsname	&INDUN..&BASE.PY(n)
DCB	Recfm=VB, Lrecl=1023, Blksize=27998
Utilization	Batch
Size	1018 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..&BASEPJ(n)
DCB	Recfm=FB, Lrecl=170, Blksize=27880
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=21600
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=24000

Characteristics	Value
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=21600
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=26000
Dsname	&INDUN..&USER.&BASE.FO(n)
Utilization	Memorize all the results of the impact analysis.

- File of entities to be analysed (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)
Reclsize	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)
Reclsize	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

Characteristics	Value
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
Reclsize	230
CI size	4,096
Key	49 (position 0)
Utilization	Batch et TP
Dsname	&INDUV..BVPGS

Complementary Libraries and Files

Complementary Dictionary files are automatically downloaded via SMP/E in the hlq.SBVPDIC PDS.

The description of these files is the following one:

PQC FUNCTION

- The BVPQCREN and BVPQCRFR members contain the standard quality rules.

- SPECIFIC MEMBERS OF PACBENCH QUALITY CONTROL FUNCTION

Member	Contents of format	Comments
BVPQCRE	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.

EXECUTION CONDITIONS

The database must be closed to on-line processing.

ABNORMAL EXECUTIONS

Refer to Chapter 'Batch Procedures', Subchapter 'Abnormal Executions' of the Administrator's Guide.

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

The VING 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

PRINTED OUTPUT

The procedure outputs:

- a report listing the executed programs,
- the list of requests with the errors detected if any,
- a report of the updates performed by the installation.

RESULT

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

NOTE

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

INPUT RECOGNITION: PTU001

INITIALIZATION OF THE KSDS WORKING FILE: IDCAMS

VSAM FILES CHECK: IDCAMS

UPDATE OF THE ADMINISTRATION DATABASE: VING

Code	Physical name	Type	Label
BVP7AE	&INDSV..BVP AE	Input	error labels
BVP7GN	&INDSV..BVP GN	Input	Administration Database Index
BVP7GR	&INDSV..BVP GR	Input	Administration Database Data
BVP7GY	&INDSV..BVP GY	Input	Administration Database Extension
BVP7GU	&INDSV..BVP GU	Input	Administration Database Users
BVP8GJ	&INDSV..BVP GJ	Input	Administration Database Journal
BVP8GN	&INDSV..BVP GN	Input	Administration Database Index
BVP8GR	&INDSV..BVP GR	Input	Administration Database Data
BVP8GY	&INDSV..BVP GY	Input	Administration Database Extension
PAC7MA	&FDIC	Input	IBM Meta-Entities Transactions
PAC7MB	&&VINGMB	Input	User Input

Code	Physical name	Type	Label
PAC7BM	&&PACXBM	Input/Output	Work file
PAC7WD	&&PACXWD	Input/Output	Extracted Transactions
PAC7ES	&&PACXES	Input/Output	Extracted Transactions
PAC7TD	&&PACXTD	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		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
- 4: Warning
- 8: No access authorization for batch procedure or invalid database (in such a case, restart the procedure with 'D' in column 16)
- 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,FMT0=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(DFSYSXEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSXEXT,DISP=(,PASS),SPACE=(TRK,1),
//          UNIT=&UWK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//-----
//DEFINE EXEC PGM=IDCAMS
/**:STEPACAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT

```

```

//SYSIN DD DSN=&&DFSYSSEXT,DISP=(OLD,DELETE)
//*-----
//MAXKEY EXEC PGM=IDCAMS
//*:STEPCHAT 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(REPR0999),DISP=SHR
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGY DD DSN=&INDSV..BVPGY,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PACGGJ DD DSN=&INDSV..BVPGJ,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGY),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGJ),DISP=SHR
//*-----
//VINS EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPVINS,GINS$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
//*:STEPCHAT 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
//*:STEP CAT 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 network using transactions provided by IBM.

Entities are created in inter-Library mode and in the 0001Z session. They can thus be accessed from any Library of the network and from any session.

EXECUTION CONDITIONS

The database must be closed to on-line processing.

ABNORMAL EXECUTIONS

Refer to Chapter 'Batch Procedures', Subchapter 'Abnormal Executions' of the Administrator's Guide.

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

The VINS 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

Position	Length	Value	Meaning
		' '	'Y' + 'R'

PRINTED OUTPUT

The procedure prints

- a report listing the executed programs,
- the list of requests with the errors detected if any,
- a report of the updates performed by the installation,
- a report of the updates performed by the retrieval,

RESULT

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

For the retrieval, a sequential file of purge transactions for the reorganization procedure is generated. After the retrieval, the reorganization of the Database is required.

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 NETWORK: 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
BVP7GY	&INDSV..BVPGY	Input	Administration Database Extension

Code	Physical name	Type	Label
BVP7GU	&INDSV..BVPGU	Input	Administration Database Users
B7AJ\$BASE	&INDUV..&BASE.AJ	Input	Development Database Journal
B7AN\$BASE	&INDUV..&BASE.AN	Input	Development Database Index
B7AR\$BASE	&INDUV..&BASE.AR	Input	Development Database Data
B7AY\$BASE	&INDUV..&BASE.AY	Input	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	&&PACXTD	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		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=$BCOB,DISP=SHR
//SYSOUT DD SYSOUT=&OUT
//PACRIN DD DSN=&INDSN..BVPSY(DFSYSEXT),DISP=SHR
//PACROU DD DSN=&&DFSYSEXT,DISP=(,PASS),SPACE=(TRK,1),
// UNIT=&UWK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//*-----
//DEFINE EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DFSYSEXT,DISP=(OLD,DELETE)
//*-----
//MAXKEY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSPAF DD DSN=&INDUV..SYSEXT.&USER,DISP=SHR
//MAXKEY DD DSN=&INDSN..BVPSY(MAXKEY),DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(REPRO999),DISP=SHR
//*-----
//VERIFY EXEC PGM=IDCAMS
//*:STEPCHAT DD DSN=&VSAMCAT,DISP=SHR
//*: DD DSN=&SYSTCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//PACGGN DD DSN=&INDSV..BVPGN,DISP=SHR
//PACGGR DD DSN=&INDSV..BVPGR,DISP=SHR
//PACGGU DD DSN=&INDSV..BVPGU,DISP=SHR
//PAC7AJ DD DSN=&INDUV..&BASE.AJ,DISP=SHR
//PAC7AN DD DSN=&INDUV..&BASE.AN,DISP=SHR
//PAC7AR DD DSN=&INDUV..&BASE.AR,DISP=SHR
//PAC7AY DD DSN=&INDUV..&BASE.AY,DISP=SHR
//SYSIN DD DSN=&INDSN..BVPSY(VERIFGN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFGU),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAJ),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAN),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAR),DISP=SHR
// DD DSN=&INDSN..BVPSY(VERIFAY),DISP=SHR
//*-----
//VINS EXEC PGM=DFSRR00,REGION=$REGSIZ,
// PARM=(DLI,BVPVINS,VINS$SUG,&BUF,
// &SPIE&TEST&EXCPVR&RST,&PRLD,
// &SRCH,&CKPTID,&MON,&LOGA,&FMTO,,,&DBRC,
// &IRLM,&BKO)
//STEPLIB DD DSN=&RESLIB,DISP=SHR
// DD DSN=&STEPLIB,DISP=SHR
// DD DSN=$BCOB,DISP=SHR
//DFSRESLB DD DSN=&RESLIB,DISP=SHR
//IMS DD DSN=&PSBLIB,DISP=SHR
// DD DSN=&DBDLIB,DISP=SHR
//*:STEPCHAT 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..BVP7AE,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..BVP7GN,DISP=SHR
//BVP7GR DD DSN=&INDSV..BVP7GR,DISP=SHR
//BVP7GU DD DSN=&INDSV..BVP7GU,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
//*:STEP CAT DD DSN=&VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DSN=&&DLSYSEXT,DISP=(OLD,DELETE)
//
```

GS/HE/ZS Files Initialization Utilities



Part Number: DEPIM001301A - 6061

Printed in U.S.A.

(1P) P/N: DEPIM001301A - 6061

