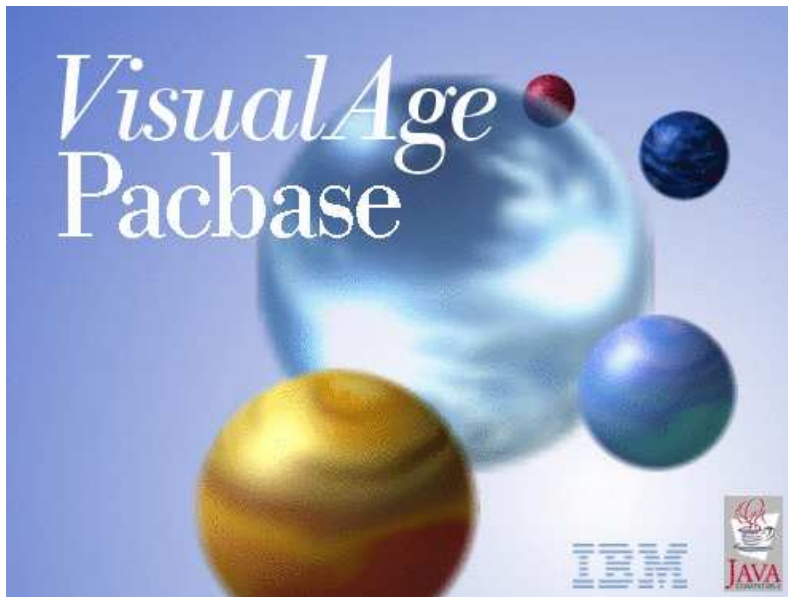


VisualAge Pacbase



# Installation Guide UNIX Server & Client Components

*Version 3.5*





VisualAge Pacbase



# Installation Guide

## UNIX Server & Client Components

*Version 3.5*

**Note**

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

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

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

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

**Second Edition (February 2016)**

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.5

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

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

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

© **Copyright IBM Corporation 1983,2016.**

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

---

# Contents

<b>Notices</b> . . . . .	<b>vii</b>	Deletion of a Database . . . . .	23
<b>Trademarks</b> . . . . .	<b>ix</b>	List of components with their date of creation . . . . .	24
<b>Chapter 1. Foreword</b> . . . . .	<b>1</b>	INSL - Execution Script . . . . .	24
Introduction . . . . .	1	<b>Chapter 4. Installation/Reinstallation of Client Components</b> . . . . .	<b>27</b>
VisualAge Pacbase Architecture . . . . .	1	Things to Know Before Installing . . . . .	27
Contents of Supply . . . . .	2	Root Directory . . . . .	27
Bibliography . . . . .	3	Installation Startup. . . . .	28
<b>Chapter 2. Prerequisites</b> . . . . .	<b>5</b>	Fundamentals of VA Pac Client-Server Communication . . . . .	28
Prerequisites for the Server Environment . . . . .	5	Administrator & Developer workbench . . . . .	30
Hardware and Software . . . . .	5	Open Jade and Tidy / Publishing facility . . . . .	33
Disk Space . . . . .	5	eBusiness Tools . . . . .	34
Prerequisites for the Client Environment . . . . .	5	VisualAge Pacbase WorkStation . . . . .	35
Hardware . . . . .	5	Web Application Models (WAM) . . . . .	41
Disk Space . . . . .	6	Middleware . . . . .	41
Software . . . . .	6	Editing Communication Parameters . . . . .	43
Communications . . . . .	6	The bases.ini File . . . . .	43
<b>Chapter 3. Installation of the Server Components</b> . . . . .	<b>7</b>	The vaplocat.ini File . . . . .	46
Installation . . . . .	7	Component Updating, Modification, or Removing. . . . .	50
General Overview . . . . .	7	<b>Chapter 5. Tests</b> . . . . .	<b>51</b>
Installation Process . . . . .	7	List of Utilities . . . . .	51
Installation of System Environment . . . . .	8	Installation Tests . . . . .	51
Update of the '/etc/services' File . . . . .	8	Generation-Print, Online and Batch Update Tests . . . . .	52
Creation of a Specific UNIX User . . . . .	9	Administration Database Procedures Tests . . . . .	52
Modification of the Configuration Files . . . . .	9	Development Database Procedures Tests . . . . .	52
Code page definition . . . . .	11	Extraction-Utility Tests . . . . .	53
Installing from a CDROM . . . . .	11	<b>Chapter 6. Server Re-installation</b> . . . . .	<b>55</b>
Description of the Installation Procedure Steps . . . . .	11	Re-installation . . . . .	55
List of installed elements. . . . .	13	General Overview . . . . .	55
An element of the System: the online server . . . . .	14	Description of the Re-Installation Procedure Steps . . . . .	55
Configuration . . . . .	14	<b>Chapter 7. Retrieval - Exchanges between 2.n &amp; 3.n Databases</b> . . . . .	<b>57</b>
Startup. . . . .	14	Retrieval of VisualAge Pacbase 2.0 and 2.5 . . . . .	57
Description of the commands . . . . .	15	Foreword . . . . .	57
Installation of the Specifications Database . . . . .	19	Notes . . . . .	57
Administration Database. . . . .	19		
Development Database . . . . .	21		
Access key entering . . . . .	23		
Backup of the Administration Database . . . . .	23		

Operations to be Performed . . . . .	58	GY Transactions exchanges between 2.n &	
Retrieval of User Parameters (PE25) . . . . .	60	3.n (GY25) . . . . .	116
PE25 - Introduction . . . . .	60	GY25 - Introduction . . . . .	116
PE25 - Input / Processing / Results . . . . .	61	GY25 - Description of Steps . . . . .	117
PE25 - Description of Steps . . . . .	62	GY25 - Execution Script . . . . .	117
PE25 - Execution Script . . . . .	64	MB Transactions exchanges between 3.n &	
Retrieval of the Development Database		2.n (MB30) . . . . .	118
(PC25) . . . . .	67	MB30 - Introduction . . . . .	118
PC25 - Introduction . . . . .	67	MB30 - Description of Steps . . . . .	119
PC25 - Notes on Data Retrieval . . . . .	67	MB30 - Execution Script . . . . .	119
PC25 - Input / Processing / Results . . . . .	69	GY Transactions exchanges between 3.n &	
PC25 - Description of Steps . . . . .	69	2.n (GY30) . . . . .	121
PC25 - Execution Script . . . . .	73	GY30 - Introduction . . . . .	121
Generation-Print Commands Retrieval		GY30 - Description of Steps . . . . .	121
(PG20) . . . . .	78	GY30 - Execution Script . . . . .	122
PG20 - Introduction . . . . .	78	Procedures - Summary Table of Changes	123
PG20 - Input / Processing / Results . . . . .	79	Retrieval of VisualAge Pacbase 3.0 . . . . .	126
PG20 - Description of Steps . . . . .	80	Operations to be Performed . . . . .	126
PG20 - Execution Script . . . . .	84		
Generation-Print Commands Retrieval		<b>Chapter 8. Components . . . . .</b>	<b>127</b>
(PG25) . . . . .	88	Server Environment Components . . . . .	127
PG25 - Introduction . . . . .	88	Introduction . . . . .	127
PG25 - Input / Processing / Results . . . . .	89	On-Line Documentation . . . . .	127
PG25 - Description of Steps . . . . .	89	Generation Skeletons . . . . .	127
PG25 - Execution Script . . . . .	93	Administration Database . . . . .	128
Retrieval of PJ Transactions (PJ25). . . . .	97	Administration Database Files . . . . .	128
PJ25 - Introduction . . . . .	97	Administration Database Backup. . . . .	128
PJ25 - Description of Steps . . . . .	98	Development Database . . . . .	128
PJ25 - Execution Script . . . . .	99	Development Database Files . . . . .	128
Retrieval of PP file (PP25) . . . . .	100	Development Database Backup Files	129
PP25 - Introduction . . . . .	100	Modules - Specific Files . . . . .	129
PP25 - Input / Processing / Results . . . . .	101	Pac/Impact . . . . .	129
PP25 - Description of Steps . . . . .	101	DSMS. . . . .	130
PP25 - Execution Script . . . . .	103	PAF . . . . .	130
Retrieval of passwords (UTMP) . . . . .	106	Complementary Libraries and Files	130
UTMP - Introduction. . . . .	106		
UTMP - Description of Steps . . . . .	107	<b>Chapter 9. Appendix . . . . .</b>	<b>133</b>
UTMP - Execution Script . . . . .	107	Installation of the Administration Database	
Retrieval of Pac/Transfer Parameters		Model . . . . .	133
(UV25) . . . . .	108	VINS - Introduction . . . . .	133
UV25 - Introduction . . . . .	108	VINS - Input / Processing / Results . . . . .	133
UV25 - Input / Processing / Results . . . . .	108	VINS - Description of Steps . . . . .	134
UV25 - Description of Steps . . . . .	109	VINS - Execution Script . . . . .	136
UV25 - Execution Script . . . . .	111	Installation of the Development Database	
MB Transactions exchanges between 2.n &		Model . . . . .	138
3.n (MB25) . . . . .	113	VINS - Introduction . . . . .	138
MB25 - Introduction . . . . .	113	VINS - Input / Processing / Results . . . . .	138
MB25 - Description of Steps . . . . .	114	VINS - Description of Steps . . . . .	140
MB25 - Execution Script . . . . .	115	VINS - Execution Script . . . . .	141
		GS/HE/ZS Files Initialization Utilities . . . . .	143

LDGS - Introduction . . . . .	143	UTU1 - Input / Processing / Results	147
LDGS - Description of Steps . . . . .	144	UTU1 - Description of Steps . . . . .	148
LDGS - Execution Script . . . . .	144	UTU1 - Execution Script . . . . .	148
LDHE - Introduction. . . . .	145	UTU2 - Update of 'P' lines with 'UNS'	149
LDHE - Description of Steps . . . . .	145	UTU2 - Input / Processing / Results	149
LDHE - Execution Script . . . . .	145	UTU2 - Description of Steps . . . . .	149
LDZS - Introduction . . . . .	146	UTU2 - Execution Script . . . . .	150
LDZS - Description of Steps . . . . .	146	UTM1 UTM2 - Conversion of 'old' Meta	
LDZS - Execution Script . . . . .	146	Entities . . . . .	151
Retrieval utilities . . . . .	147	UTM1 - Description of Steps . . . . .	153
UTU1 UTU2 - Adaptation of 'UNS'		UTM1 - Execution Script . . . . .	154
operators . . . . .	147	UTM2 - Input / Processing / Results	155
UTU1 - Extraction of 'P' lines with		UTM2 - Description of Steps . . . . .	155
'UNS'. . . . .	147	UTM2 - Execution Script . . . . .	156





---

## Notices

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

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

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

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



---

## Trademarks

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

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

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

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

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



---

# Chapter 1. Foreword

---

## Introduction

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

- Server components
- Client components,
- Communication environment.

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

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

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

The procedures that migrate the Pacbase data to Rational Programming Patterns are documented in the "Migration" entry of the Rational Programming Patterns Knowledge Center.

---

## VisualAge Pacbase Architecture

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

VisualAge Pacbase consists of:

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

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

**NOTE:** The textual mode remains available for some functionalities.

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

## The Server environment

It consists of the following components:

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

## The Client environment

The client environment includes five separately installable components:

- Administrator & Developer workbench (also separately installable).  
Developer workbench includes the following modules (each running independently):
  - Batch,
  - Dialog,
  - eBusiness (includes three eBusiness Tools and Services Modeler),
  - Services Modeler.

Each of these modules can be executed independently.

- eBusiness Tools:
  - Proxy Generator,
  - Location Editor,
  - Services Test Facility,
  - VisualAge Pacbase Connector,
  - VisualAge Pacbase Web Connection (Dialog Web Revamping Generator),
  - Web application models (WAM).
- VisualAge Pacbase WorkStation (Pacdesign, Pacbench)
- Web application models (WAM)
- Middleware

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

---

## **Contents of Supply**

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

- Installation guide,
- Server installation CD-ROM,
- Client installation CD-ROM,
- Documentation CD-ROM.

---

## Bibliography

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





---

## Chapter 2. Prerequisites

---

### Prerequisites for the Server Environment

#### Hardware and Software

- Architecture: A UNIX server
- Memory: RAM memory (64 Mb). You may need additional memory according to the number of servers installed on the same machine.
- Software - COBOL runtime
  - For the Micro Focus version:  
Micro Focus Server for COBOL 5.0 or greater.  
VA Pacbase is available on a 32-bit version on the AIX, HP, Linux, and Sun platforms.  
On the AIX and HP platforms, a 64-bit version is also available.
  - For the ACUCOBOL-GT version:  
ACUCOBOL-GT 7.2 or Extend 8.  
VA Pacbase is available on 32 and 64-bit versions on the AIX platform.
- Installation medium: CD-ROM drive.

#### Disk Space

The disk space occupied by files varies according to the size of the applications managed by the system.

The table here below indicates approximately (in million bytes) the disk space required by the installation of servers:

Size of installation	Total
Total for installation	140
Total for system	116
Total for installation tests user files	22
Total for installation tests administrator files	2

---

### Prerequisites for the 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): 512 Mb (1 Gb advised).

## **Disk Space**

Required disk space:

- 100 Mb for Administrator & Developer workbench
- 30 Mb for eBusiness Tools
- 85 Mb for the "IBM SDK for Java 2"
- 25 Mb for the Middleware
- 6 Mb for WAM
- 30 Mb for the VisualAge Pacbase WorkStation (with a Methodology included)

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

For an operational installation of AD workbench for example, the minimum disk space required is 215 Mb (100 Mb for AD workbench, 85 Mb for the "IBM SDK for Java 2" and 30 Mb for the Middleware).

## **Software**

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

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

---

## **Communications**

To enable the communication between the workstation components and the servers in a UNIX environment, the communication protocol is TCP-IP Socket.

---

## Chapter 3. Installation of the Server Components

---

### Installation

#### General Overview

The product is supplied on a CDROM which contains:

- pacinst.Vnn,
- PACBASE.xxx.

('Vnn' represents the number of the installed version).

('xxx' represents the COBOL runtime version).

**WARNING:** Depending on the support and the UNIX system, the names of the installation files might be in upper case letters or in lower case letters.

#### Description of pacinst.Vnn :

The pacinst.Vnn file contains the VA Pac installation or reinstallation commands.

These commands are written in shell language.

#### Description of PACBASE.xxx:

The PACBASE.xxx files are compressed tar files which contain the VA Pac execution and operations files.

When installing or reinstalling, you can choose the runtime and then install files which are compatible with the runtime in use.

#### Installation Process

It is recommended to backup the system before starting the VA Pac installation and to closely follow the various installation steps.

- Update of the '/etc/services' file,
- Creation of a specific login to the software,
- Creation or modification of the environment variables,
- installation of the system, from the CD-ROM,
- installation of the Administration Database,
- installation of the Development Database,

Between the system installation and the Databases installation, the Mklink is executed. If its execution abnormally ends, you just have to correct the error and link the executable programs to the runtime if needed. Then, restart the installation of the Databases by executing the "pacadmin" procedure (see the "Installation of the Specifications Database" section in this chapter) stored in the following directory: \$PACDIR/system/install.

**WARNING:** VA Pac must be installed on disks physically present on the UNIX machine and not on the NFS-mounted disks.

For the Microfocus or Acucorp versions, the COBOL runtime is required for VA Pac to operate properly. It must be installed before VA Pac.

**WARNING:** For the installation of the Acucobol version on HP-UX, you must assign the [nflocks] system variable, 'max Number of File LOCKS', a value higher than 9000.

---

## Installation of System Environment

### Update of the '/etc/services' File

The use of the 'socket' communication interface for the communication between the server and the clients requires the reservation of communication port numbers in the '/etc/services' UNIX file.

This file must be updated BEFORE the installation of the Development Databases. Since this installation can follow the system installation, this file must be updated BEFORE the system installation.

A communication port number must be associated with each database. Communication ports cannot be chosen at random.

Some ports are reserved for standard 'internet' applications or for later applications. Other ports are allocated in a dynamic way.

So you must choose a port number which is not used in '/etc/services' and which is not being used, i.e. dynamically allocated.

Example:

A port number must be chosen for each on-line server, including the ADMIN Database. Let us suppose that we have chosen number 52030.

- The port number 52030/tcp must not be in the '/etc/services' file ;

- The port number must not be currently used. You can make sure of that via the following UNIX command:

```
netstat -an | grep 52030
```

- If both these conditions are met, the UNIX (root) administrator must update the '/etc/services' file using the following line:

```
paclanx 52030/tcp
```

The port number chosen for each on-line server must be carefully noted because it will be needed when creating Development Databases.

## **Creation of a Specific UNIX User**

Most software running on UNIX must be installed in a specific user's account. This is necessary for security reasons, in order to isolate the various software programs installed on one machine.

NEVER INSTALL UNDER THE "ROOT" UNIX ACCOUNT!

The UNIX administrator (root) must therefore create a UNIX user under which VA Pac will be installed.

This user will be the VA Pac administrator's.

He must be authorized to use the 'at' command.

## **Modification of the Configuration Files**

Log in with the VA Pac administrator login, and update the shell configuration files (.profile, .kshrc or .login).

The environment variables PACDIR and PATH will be created or modified:

```
PACDIR=$HOME/paclanx
export PACDIR
PATH=$PACDIR/system/bin:$PACDIR/system/proc
:$COBDIR/bin:$PATH
export PATH
```

The environment variables COBDIR, COBPATH, ACUDIR and, CODE\_PREFIX must be added, according to the runtime:

For Microfocus :

```
COBDIR=<COBOL runtime directory path>
export COBDIR
COBPATH=$PACDIR/system/gnt
export COBPATH
```

For ACUCOBOL -GT:

```
ACUDIR=<COBOL runtime directory path>
export ACUDIR
CODE_PREFIX=$PACDIR/system/acu
export CODE_PREFIX
```

If the TMPDIR variable (temporary directory of the COBOL runtime) is assigned, the indicated directory must exist.

For the AIX version of Microfocus, you must create or modify the LIBPATH variable:

```
LIBPATH=/usr/lib:$COBDIR/coblib:$PACDIR/system/bin
export LIBPATH
```

If you use ACUCOBOL -GT, LIBPATH must contain:

```
LIBPATH=/usr/lib:$ACUDIR/lib:$PACDIR/system/bin
export LIBPATH
```

For the HP, Linux or Sun versions of Micro Focus, you must use LD\_LIBRARY\_PATH instead of LIBPATH.

For the HP version, in addition to the LD\_LIBRARY\_PATH variable, you must set the SHLIB\_PATH variable as follows:

```
export SHLIB_PATH=$PACDIR/system/bin:$SHLIB_PATH
```

The COBOL runtime corresponds to the directory which contains the executable files and libraries required to execute COBOL programs.

For Micro Focus, COBDIR can generally take the following values: /usr/lib/cobol, /usr/lpp/cobol or /opt/cobol. (For further details on the COBDIR environment variable, refer to the installation documentation of your COBOL compiler and to your COBOL manuals.)

For ACUCOBOL -GT, the ACUDIR variable contains the compiler or the COBOL runtime installation directory.

Once you have performed these updates, restart the configuration initialization by typing '.profile', and use the UNIX command 'set' to check that the modifications were taken into account in the configuration files.

If the environment variables are not correctly initialized, log out and then log in again, to the VisualAge Pacbase administrator's account.

## Code page definition

The Repository code page is an IBM-923 code page equivalent to the ISO8859-1 code page which supports the Euro character (ISO8859-15).

### Positioning the 'LANG' code for the UNIX user

In order to visualize properly the reports which contain national characters, the user must define a code page compatible with the VA Pac Repository code page.

Refer to your documentation related to the UNIX system to define a code page compatible with the ISO8859-1 or ISO8859-15 code page.

As a general rule, the positioning of the 'LANG' variable alone is sufficient to define the code page to be used.

Example : for the support of French national characters, you set:

- on AIX, SOLARIS:  
`export LANG=fr_FR.ISO8859-1`
- on HP-UX :  
`export LANG=fr_FR.iso8859-1`

## Installing from a CDROM

**WARNING:** If necessary, mount the CD-Rom disk driver on a system directory.

## Description of the Installation Procedure Steps

To ensure a correct installation of the software, check that the disk space is equal to five times the size of the PACBASE.xxx file in the installation directory.

After unloading in the VA Pac administrator's login directory, or the CDROM mounting, go to the following directory:

```
PB350"package_code"."version_code"
```

which is located in \$HOME or in the CDROM mounting directory.

Example on AIX:

```
cd $HOME/PB350AIX.Vnn (nn = version number)
```

and type the following command:

```
sh pacinst.Vnn (nn = version number)
```

followed by a carriage-return (Enter or Return key).

The installation is made up of the following steps:

- Consistency check on the environment variables,
- Display of the installation menu
- Selection of the COBOL runtime in use
- Selection of the generation language code
- Creation of the installation directory
- Distribution of files coming from PACBASE.xxx
- Creation of the journal\_pacinstall file
- Creation of the Administration Database
- Creation of the standard test Database (BVAP).

### Consistency check of the environment variables

Before running pacinst.Vnn, the VA Pac Administrator must check the modifications done in the configuration file, even though the installation process performs some consistency checks.

### Display of the installation menu

The installation menu looks like this:

```
*****  
                                VisualAge Pacbase  
*****  
inst          : version installation  
*****  
Type the command name or 'x' to exit pacinst.Vnn :
```

Description of commands:

- The 'inst' command starts the installation process.

Enter the command and follow the instructions given by pacinst.Vnn.

### Choice of the COBOL runtime in use

This choice is proposed on some platforms at installation.

Ask your UNIX Administrator to know which runtime is used.

### Selection of the generation language

The generation language code is 'en' for English or 'fr' for French.



### Creation of the installation directory

pacinst.Vnn requires the installation directory to be named paclanx in order to:

- isolate VA Pac from the other applications and so control its evolution more easily,
- allow pacinst.Vnn to control the type of processing (installation or reinstallation) to be done.

### Distribution of files:

During this step, pacinst.Vnn creates the directories described further on in this manual (paragraph 'Description of created Directories') and extracts all the PACBASE.xxx files.

### Update of the journal\_pacinstall file:

The journal\_pacinstall file contains information resulting from the use of pacinst.Vnn. During the installation process, this information is:

- the installation date,
- the release number,
- the version number,
- the generation language code.

### Note:

The BVPDS600 and BVPDS610 load-modules for the interface between VA Pacbase and DSMS 2.5 are provided in standard; for a DSMS use, the DEXP procedure must be adapted to use these new load-modules instead of PDS600 and PDS610 which have no longer been compatible since VA Pacbase 3.0.

## **List of installed elements**

The installation copies:

- the translation file, the labels file produced by the procedures in case of error or for information,
- the skeletons in the languages supported by the product,
- the programs,
- the procedures,
- the Database creation utility,
- the startup script models.

## An element of the System: the online server

### Configuration

When started, each listener executes a 'BVPSEVER.ini' command file type to assign any environment file it needs.

This file is created when the listener is created in the \$PACDIR/config/"database code" directory and listener environment variables:

- BVPSOCKET : socket number,
- SRV\_DIR : directory containing the listener trace files,
- SEMLOCK : serialization lock of conflicting updates of the user Database,
- SEMADMIN : serialization lock of conflicting updates of the 'ADMI' administrator Database.

### Startup

The listener, whose executables (pactp, bvpserver, paclaunch) are in the \$PACDIR/system/bin directory allows to:

- Set the listener(s) in active or inactive mode,
- Supply information concerning the listener(s),
- Purge one or more workstation attached to a listener,
- Purge the listener(s).

To carry out the operations listed above, the listener commands interpreter (pactp) is started up.

There are two possible modes:

- the 'command' mode,
- the 'shell' mode.

For the 'command' mode type, enter:

```
pactp <command>
```

To return to 'shell' mode, enter:

```
pactp -s
```

The 'command' mode is used to insert the listener commands in a commands file.

For example, the com\_paclan file contains the following commands:

```

# display the listener' status
pactp info
# start the BVAP listener
pactp start BVAP
# display information on the BVAP listener in ten seconds
pactp info BVAP
# stop the ADMI listener
pactp shutdown ADMI
# start the 003 station purge command dependant on the BVAP
listener
pactp purge BVAP 003

```

The advantage of the 'shell' mode is that you avoid re-starting the interpreter for each command and so you avoid delays between commands.

The commands available are:

- debug (activation/deactivation of debugger mode)
- exit (exit from 'shell' mode),
- help (help on a command's syntax),
- info (information on the listener(s)),
- purge (purge the workstation),
- purge\_server (purge the listener),
- shutdown (stop the listener without confirmation),
- start (start the listener),
- stop (stop the listener with confirmation),
- view (display the listener's status).

A detailed description of the commands is given below in the paragraph 'Description of the commands'.

### Execution conditions

The PACDIR environment variable must be initialized.

The COBPATH environment variable must contain the access path to the online modules: '\$PACDIR/system/gnt'.

(see chapter 'Installation of Server Environment' - 'Installation of System Environment' - 'Modification of the Configuration Files').

## **Description of the commands**

### Debug command

This command activates or deactivates the 'debug' mode. The listener name must be passed as a parameter, followed by 'on' or 'off' to activate or deactivate the 'debug' mode.

There are two types of result files:

```
srv[process_number].txt
```

to trace the processing of the listener which listens new connections.

```
dial[process_number].txt
```

to trace the processing of each connection to the listener.

So there is a trace for each connection to the listener.

These files are located in the directory pointed by the SRV\_DIR environment variable. This variable is assigned in the \$PACDIR/config/"database code"/BVPSERVER.ini configuration file. Its default value is: \$PACDIR/data/"database code"/tmp/server.

EXAMPLE:

- Start the debug mode on the BVAP listener

```
pactp debug BVAP on      (in command mode)
VA Pac  : debug BVAP on  (in shell mode)
```
- Stop the debug mode on the BVAP listener

```
pactp debug BVAP off    (in command mode)
VA Pac  : debug BVAP off (in shell mode)
```

- Different trace levels can be applied:

1. Level 1

Minimum trace to follow the listener processing with the calls to the COBOL communication monitor,

2. Level 2

Detailed trace of the listener processing,

3. Level 4

Trace of messages exchanged between the listener and the client.

The 'debug on' command activates a level 1 trace on the active listener. To use other traces, the listener must be restarted after setting the SRV\_TRACE variable in the 'BVPSERVER.ini' file.

EXAMPLE:

SRV\_TRACE=1 for a level 1 trace

SRV\_TRACE=3 for a level 1 and 2 trace

SRV\_TRACE=5 for a level 1 and 4 trace

The 'debug off' command stops the trace creation for the new listener connections.

### Exit command

This command is used to exit the shell mode (commands interpreter) previously activated by pactp -s.

### Help command

This command displays help concerning a listener administration command. If no parameter is specified, the list of available commands is displayed. If a command is specified, the help for this command is displayed on the screen.

EXAMPLE: request help on the syntax of the start command

```
start
pactp help start      (in command mode)
VA Pac  : help start  (in shell mode)
```

### Info command

This command displays information on the listener(s).

This command followed by a base code (info "database code") gives the following information :

- active (if the listener is active),
- inactive (is the listener is inactive),
- Error (if the listener is stopped abnormally).

### Purge command

This command purges a workstation (i.e. it disconnects a workstation).

If the interpreter (pactp) is in command mode, the syntax for the purge command is as follows:

```
'pactp purge <database code> <no. of the station to be purged>'
```

In shell mode there are two syntaxes for the purge command:

- 'purge <no. of the station to be purged>' if the prompt is different from 'VA Pac :' (the prompt taking the value of listener name, after the use of the view command, for example).
- 'purge <database code> <no. of the station to be purged>' if the prompt is 'VA Pac :'.

### Purge\_server command

This command purges a listener in the case of abend, displayed with the 'Error' status after the 'info' command. The syntax for the purge\_server command is as follows:

```
'pactp purge_server <database code>'
```

In shell mode the syntax for the command is:

```
'purge_server <database code>'
```

N.B.: This command deletes the listener process(es), as well as the used IPC resources (semaphores, shared memory).

### Shutdown command

This command stops the listener. The database code must be passed as a parameter of the command.

EXAMPLE: stop the BVAP listener

```
pactp shutdown BVAP      (in command mode)
VA Pac : shutdown BVAP  (in shell mode)
```

### Start command

This command starts the listener. The database code must be passed as a parameter of the start command. The initialization file BVPSEVER.ini must be in the \$PACDIR/config/'database code' directory. The listener start-up program, paclaunch, starts the 'bvpsver' process. If problems arise (listener lock), use the following command first:

```
pactp purge_server 'database code'.
```

If this command has no effect, the 'bvpsrver' process can always be killed using the 'kill -15' command or 'kill -9' followed by process number (PID).

EXAMPLE: starting the BVAP listener

```
pactp start BVAP      (in command mode)
VA Pac : start BVAP  (in shell mode)
```

### Stop command

This command stops a listener. The database code must be passed as a parameter of the stop command. You have to confirm stopping the server.

EXAMPLE: stop the BVAP listener

```
pactp stop BVAP      (in command mode)
VA Pac : stop BVAP  (in shell mode)
```

### View command

This command displays information concerning a listener: the list of the connected workstations, the IP address of the client workstation, the PID number of the client process launched by the listener, the VA Pac user code, the name of the executed COBOL program and its 'Elapse' execution time in milliseconds. It also gives information on the status of the semaphore used to synchronize the concurrent accesses to the database.

EXAMPLE: display information about the BVAP listener

```
pactp view BVAP      (in command mode)
VA Pac : view BVAP  (in shell mode)
```

---

## **Installation of the Specifications Database**

### **Administration Database**

This step creates the Administration Database which controls the Development Databases. Only one Administration Database can be installed. Its code is 'ADMI'.

The Administration Database must be installed right after the system installation, or later with the 'pacadmin' procedure located in the directory:

\$PACDIR/system/install.

This installation is constituted of the following steps:

### Display of the installation menu

The installation menu looks like this:

```
*****
                          VisualAge Pacbase
*****
  creadmin      :  creation of the administration database
*****
Type the command name or 'x' to exit pacadmin :
```

Description of commands:

- The 'creadmin' command starts up the installation process.

Enter the command and follow the instructions given by pacadmin.

#### Input the language code in the Database

- en (English)
- fr (French)

#### Input the socket number of the listener

The socket number must be between 49152 and 65535.

It must be declared in the /etc/services file (refer to 'Update of the /etc/services File' sub-chapter).

#### Creation of the Database subdirectories

- \$PACDIR/data/ADMI/base
- \$PACDIR/data/ADMI/journal
- \$PACDIR/data/ADMI/save
- \$PACDIR/data/ADMI/script
- \$PACDIR/data/ADMI/tmp
- \$PACDIR/data/ADMI/users
- \$PACDIR/config/ADMI

#### Execution of the initialization batch procedures

- INGU : initialization of GU file,
- INGK : initialization of GK file,
- INQJ : initialization of QJ file,
- REST : Database restoration,
- VINS : initialization of the Metamodel.

#### INGU : Initialization of the Users file



This procedure is executed for the first installation only. It defines and then loads the GU file with the 'TEST' user code (blank password) and 'ADMIN' user code ('ADMIN' password). These user codes will be used to activate the keys and start the test scripts.

### Update of the journal\_pacinstall file

The journal\_pacinstall file also includes information related to the Database creation:

- Database name and creation date.

## **Development Database**

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

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

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

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

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

More than one Development Database can be installed ; each has its own environment.

A Development Database can be installed right after the system installation, or later, with the 'pacadmin' procedure which is located in the directory:

```
$PACDIR/system/install.
```

This installation involves the following steps:

### Display of the installation menu

The installation menu looks like this:

```
*****  
                                VisualAge Pacbase  
*****  
crebase      : creation of a new database
```

\*\*\*\*\*

Type the command name or 'x' to exit pacadmin :

Description of commands:

- The 'crebase' command starts up the installation process.

Enter the command and follow the instructions given by pacadmin.

#### Input of the Database code (except for the test Database)

Enter the Database code on 4 alphanumeric uppercase characters and its label on 30 alphanumeric uppercase characters.

This code must have been declared in the Administration Database.

When installing the first Database (test Database), the Database code is always BVAP.

#### Selection of the Database language code

This code can be:

- en (English)
- fr (French)

#### Input of the socket number allocated to the listener

The socket number is a 5-digit number between 49152 and 65535.

You must specify this number manually in the vaplocat.ini file (see the 'COMMUNICATION FILES' in chapter 4).

It must be declared in the /etc/services file. (see paragraph 'Update of the "/etc/services" File').

#### Creation of the Database subdirectories

- \$PACDIR/data/BVAP/base
- \$PACDIR/data/BVAP/journal
- \$PACDIR/data/BVAP/save
- \$PACDIR/data/BVAP/script
- \$PACDIR/data/BVAP/tmp
- \$PACDIR/data/BVAP/users
- \$PACDIR/config/BVAP

### Execution of the initialization batch procedures

- UPGP : update of the Development Database code and label in the Administration Database,
- REST : loading of the test Database,
- VINS : initialization of the metamodel,
- LDHE : initialization of HE file,
- LDZS : initialization of ZS file,
- LDGS : initialization of GS file,
- INFQ : initialization of FQ file,
- INFP : initialization of FP file,
- INFO : initialization of FO file.

### Update of the journal\_pacinstall file

The journal\_pacinstall file also includes the information related to the Database creation:

- Database name and creation date.

### **Access key entering**

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

- Entering the access key,
- Defining generation targets,
- Activating the key.

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

### **Backup of the Administration Database**

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

### **Deletion of a Database**

To delete a Database, you must be positioned in the installation user connection directory and delete the following directories:

`$PACDIR/data/"database code"`

and

`$PACDIR/config/"database code"`

Use the command: `rm -r [directory_name]`

## List of components with their date of creation

### INSL procedure

This procedure is used to edit:

- the list of batch, online programs and their generation characteristics,
- In case of system operation problem, this list should be printed in order to communicate to VisualAge Pacbase Support all the installation references.

In any case, you must verify that the provided user input is adapted to your environment.

### **INSL - Execution Script**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) INSL BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     - LIST OF INSTALLED PROGRAMS AND FILES -
# *
# * -----
# *
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "INSL"
echo "          ====="
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
echo ""
OUTFILE=`BVPENV INSL OUTFILE $PACUSERS/INSL.txt`
for i in `cat $PACINPUT`
do
  if [ -d "$i" ]
  then
    cd $i
    ls * >> $PACTMP/LIST
  else
    echo $i >> $PACTMP/LIST
  fi
done
```

```
fi
done
LIST=$PACTMP/LIST
export LIST
LOGFILE=$PACUSERS/INSL.log
export LOGFILE
BVPMSG 1009 "bvupdate"
bvupdate $OUTFILE > $LOGFILE 2>&1
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "bvupdate"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN
```



---

## Chapter 4. Installation/Reinstallation of Client Components

---

### Things to Know Before Installing

- To install the VisualAge Pacbase Client components on a Windows workstation, you must have an Administrator profile.
- VA Pac Client components are installed via InstallShield for Windows Installer (ISWI).  
If Windows Installer is not installed on the workstation, it will be installed automatically.
- You also need Microsoft Windows Script, version 5.1 or higher. You can download it from the following URL:  
<http://www.microsoft.com/downloads>
- Both the Administrator & Developer workbench and the eBusiness Tools components require, for their online help, that a standard internet browser be used (Mozilla Firefox or Internet Explorer is advised).
- The installation of a Client component does not require the prior installation on the server of the VA Pac Database(s) to which it will connect. However, the code of each VA Pac Database you must indicate when you install some Client components will have to be strictly reused when these Databases are installed at the server level.

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

### Root Directory

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

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

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

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

**NOTE:** Do not use blank characters in directory names.

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

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

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

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

## **Installation Startup**

Insert the installation CD-Rom.

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

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

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

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

Choose the Client component you want to install.

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

---

## **Fundamentals of VA Pac Client-Server Communication**

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

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

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

AVAILABLE COMMUNICATION TYPES



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

## MIDDLEWARE

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

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

However you can also choose a communication via a gateway.

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

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

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

## COMMUNICATION FILES

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

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

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

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

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

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

#### THE VAPLOCAT.INI FILE

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

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

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

A location :

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

#### THE BASES.INI FILE

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

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

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

---

## Administrator & Developer workbench

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

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

- Administrator workbench
- Developer workbench

Developer workbench includes the following modules, each running independently:

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

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

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

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

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

- If you choose the direct communication option, the middleware installation script will automatically start once the installation of the workbench is finished. It will ask you to specify a number of communication parameters. For complete details on this part of the installation, refer to the Middleware subchapter.
- If you choose the gateway option, enter first the IP address of the gateway here. The installation of the middleware on the Developer workstation - also necessary in this context - will then start automatically after the installation of the workbench.

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

To do this, a window enables you to enter:

1. The name of a VA Pac Database, already installed at the server level or not. The names entered here will be displayed in the connection smartguide, thus showing users which Databases they can connect to. The

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

2. The Database logical code.

Maximum length: 4 characters.

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

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

3. The location name

Maximum length: 20 characters.

Default: Location-1

More than one Database may be associated with one location.

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

You select the mode via two check boxes.

VisualAge Pacbase signon:

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

Middleware signon:

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

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

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

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

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

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

### START-UP FILES

The start-up files are :

- wb\_admin.bat
- wb\_global.bat
- wb\_batch.bat
- wb\_dialog.bat
- wb\_eBusiness.bat
- wb\_services.bat
- wb\_extensibility.bat
- wb\_cfm.bat

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

#### START MENU / PROGRAMS CHOICE

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

Administrator-Developer workbench

- Administration
- Batch
- Dialog
- eBusiness
- Services Modeler
- Extensibility
- cfm
- Global

### **Open Jade and Tidy / Publishing facility**

Since version 3.5 V05, the Publishing facility has used the XSL technology by default. So the OpenJade and Tidy tools are no longer needed.

However if you prefer to use the DSSSL publishing, you must install these products.

For the versions earlier than 3.5 V05, you must install OpenJade and Tidy to publish documents.

To obtain these tools, refer to the installation instructions on the VA Pac support internet page: <http://www.ibm.com/software/awdtools/vapacbase/download.html> .

---

## eBusiness Tools

The eBusiness Tools are:

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

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

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

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

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

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

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

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

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

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

### START-UP FILES

The start-up files are :

- For the Proxy Generator:

vapGen.exe

- For the Location Editor:

vapLocationEditor.exe

- For the Services Test Facility:

vapServicesTestFacility.exe

- For VisualAge Pacbase Web Connection:

PacWebgen.exe

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

ebusinesstools\bin

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

### START MENU / PROGRAMS CHOICE

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

eBusinessTools

Location Editor

Proxy Generator

Services test Facility

Dialog Web Revamping Generator

---

## **VisualAge Pacbase WorkStation**

**WARNING:** The WorkStation runs on a 16 or 32 bits environment. Check that the WorkStation is installed in an environment compatible with such applications.

The root used for this installation depends on this installation 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 is given at the beginning of this chapter.

This panel is not displayed 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 of the installation, see the subchapter Middleware.
- If you choose the gateway option, enter the IP address of the gateway here. The installation of the middleware on the Developer workstation - also necessary in this context - will then start automatically after the installation of the WorkStation.

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

To do this, a panel enables you to enter:

1. The name of a Database, already installed at the server level or not.  
The names entered here will be displayed in the connection smartguide, thus showing users which Databases they can connect to.



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

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

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

2. The Database logical code.

Maximum length: 4 characters.

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

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

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

3. The location name

Maximum length: 20 characters.

Default: Location-1

More than one Database may be associated with one location.

4. Finally, specify the user authentication mode upon connection. Refer to the Database Administrator to make sure that the authentication mode applied at the server level is also applied at the WorkStation level.

You select the mode via two check boxes:

'Pacbase is secured':

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

'Middleware is secured':

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

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

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

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

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

### START-UP FILE

The start-up file is :

pexec.exe

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

### START MENU / PROGRAMS CHOICE

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

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

### INSTALLATION PARAMETERS FILE

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

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

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

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

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

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

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

### DESCRIPTION OF THE PARAMETERS FILE

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

- a three-digit identifier in positions 1 to 3
- the parameter label, whose position is unfixed
- the parameter value, between square brackets ([ and ]), whose position is also unfixed

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

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

The Pacbase.dat file should not be destroyed.

The possible values for the Methodology parameter are:

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

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

### WARNING ON VAPLOCAT.INI FILE

The Vaplocat.ini file indicates the locations of the Databases and the communication parameters. If you modify this file while using the WorkStation, you must reboot your computer to take the modification into account.

For more details on this file, read the information developed in subchapters 'Things to Know Before Installing' and 'Editing communication Parameters'.

#### FILES REQUIRED FOR THE USE OF THE GENERATION-PRINT MONITOR

To activate the "Generation and Print Monitor" function in the WorkStation, the path to the Databases accessed by the user must be entered in the GPMON.PRM file.

This file is saved under the following path:

SPAC\NNNL\GPMON.PRM

If this file does not exist, you must create it in the directories indicated above. You just have to give it the name GPMON and to enter a .PRM extension.

Then, open the file, under Notepad for example, and enter a SU line for each Database accessed by the user as follows:

SU bbbb u:/Root/USERS,

bbbb for the Database code,

u for the root driver,

Root for the path to the USERS directory.

The user of the WorkStation must be able to view and access the USERS directory from his computer. To make it possible, you must use the File Share function or a network unity which should be accessible from the two stations.

In the GPMON.PRM file, you must enter the path to the USERS directory as accessible from the user's computer.

#### FILE OF NON-STANDARD LABELS

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

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

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

---

## Web Application Models (WAM)

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

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

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

---

## Middleware

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

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

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

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

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

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

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

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

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

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

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

#### LIST OF PARAMETERS

- VAP SOCKET
  - IP address: IP address and port used by the VA Pac server.  
Do not enter left-aligned '0', source of error with some configurations.
- MVS CICS SOCKET
  - IP address: IP address and port used by the VA Pac server.  
Do not enter left aligned '0', source of error with some configurations.
  - Transaction code: Code of the CICS transaction of the VA Pac Communication Monitor.
  - Code Page: Value identifying the coding of characters used by the VA Pac server.  
1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MVS CICS CPI-C
  - Destination-id entry: BVPSCPI (default value).  
If you modify this value, it must be the same as the value entered in the Symbolic destination name, a parameter included in the configuration of this communication protocol.
  - Code page: Value identifying the coding of characters used by the VA Pac server.  
1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MQ-CICS Bridge and MQ-IMS Bridge
  - LocCcsid: Character-set code of the machine in local (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.  
Maximum: 9 characters.
  - Queue Manager name: Name of the local Queue Manager of the client part.  
If there is an intermediate MQSeries Server on NT, this parameter must identify the intermediate NT Queue Manager and not the Queue Manager.  
Maximum: 4 characters.

- Request Queue name:  
Maximum: 48 characters.
- Reply Queue name:  
Maximum: 48 characters.
- Transaction code:  
Maximum of 4 characters for CICS.  
Maximum of 8 characters for IMS.
- MVS IMS Connect
  - IP address: IP address and port used by the VA Pacbase server.
  - Transaction code: IMS transaction code of the VA Pacbase Communications Monitor.
  - Code page: Value identifying the coding of characters used by the VA Pacbase server.  
1140 (US EBCDIC) or 1146 (UK EBCDIC)
  - Data Store:  
Name of the link to IMS defined in IMS Connect (IMS Data Store ID).
  - RACF group: Name of the RACF group for IMS Connect.

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

---

## Editing Communication Parameters

### The bases.ini File

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

By default this file's access path is:

C:\Program Files\IBM\VisualAge\_Pacbase\_35\Common\

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

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

The name of each section will be presented to the user in the connection smartguide. In the displayed list of VA Pac Databases, the user picks the Database he/she wants to connect to. This is why section names need be very explicit. All the more so since you can manage several communication options

for one VA Pac Database. To do so, define as many configurations/sections as needed for one Database, clearly differentiated from one another by their name.

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

## DESCRIPTION OF A SECTION'S CONTENTS

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

- baseCode = code of the VisualAge Pacbase Database (required)  
Maximum length: 4 characters

**NOTE:** Concerning the VA Pac WorkStation, this code must be unique in the bases.ini file for a given Location.

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

The following parameters vary according to the chosen option.

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

Parameter needed for the VisualAge Pacbase Workstation only.

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

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



### PARAMETERS FOR DIRECT ADAPTER (LOCAL MIDDLEWARE)

- `locationsFile` = indicates the path and name of the file which contains the locations definitions.  
Default: `..\common\vaplocat.ini`

**CAUTION:** The default value of this parameter should not be modified.

- `location` = location name for the Database  
Maximum length: 20 characters.  
Default: Location-1  
More than one Database can point to the same location.  
REMINDER: A location identifies the communication protocol used to access the VisualAge Pacbase server and the physical address of this server for this protocol.
- `traceFile` = path and name of the file which will receive the trace of the middleware execution.  
By default this file is automatically created (with timestamp) in the VapTrace sub-directory.
- `traceLevel` = trace level of the middleware execution. Its possible values are:
  - 0 : no trace
  - 1 : trace with errors (default)
  - 2 : standard trace, not detailed
  - 3 : trace for information
  - 4 and + : trace for debugging
- `codePageFile` = path and name of the file which contains the conversion table of the code pages.  
Default: `..\middleware\CharConv.txt`

### PARAMETERS FOR GATEWAY ADAPTER (REMOTE MIDDLEWARE)

- `host` = name or IP address of the host where the VisualAge Pacbase gateway is installed.  
Do not enter left aligned '0', source of error with some configurations.  
Default: 127.0.0.1 for a local host
- `port` = value of the IP port where the gateway receives the client requests.  
Default: 5647
- `location` = location name for the Database  
Maximum length: 20 characters.  
Default : Location-1  
More than one Database can point to the same location.

REMINDER: A location identifies the communication protocol used to access the VisualAge Pacbase server and the physical address of this server for this protocol.

- The traceFile and traceLevel parameters can be used to start the GATEWAY. For more details, refer to the Middleware User's Guide.

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

## The vaplocat.ini File

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

By default, the path to this file is:

C:\Program Files\IBM\VisualAge\_Pacbase\_35\Common\

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

The maximum length of this name is 20 characters.

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

- VAP SOCKET

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

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

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

- MQ-CICS Bridge

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

- IMS Connect

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

- MQ-IMS Bridge

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

## DETAILS ON THE PARAMETERS

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

- COMM\_TYPE:

This parameter identifies the communication protocol in use.

The possible values are:

- SOCKET: VA Pac Server under Windows or UNIX, with the use of TCP/IP.
- TCPMVS: VA Pac Server under MVS/CICS with the use of a TCP/IP listener.
- CPIC: VA Pac Server under MVS/CICS, with the use of the CPI-C protocol.
- MQCICS: VA Pac Server under MVS/CICS, with the use of the MQ-CICS-BRIDGE protocol.
- MQMCICS: Same as MQCICS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.
- TCPIMS: VA Pac Server under MVS/IMS, with the use of the IMS Connect protocol.
- MQIMS: VA Pac Server under MVS/IMS, with the use of the MQ-IMS-BRIDGE protocol.
- MQMIMS: Same as MQIMS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.

- IXO\_ADDRESS: IP address and port used by the VA Pac Server.

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

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

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

- IXO\_DATASTORE: Name of link to IMS defined in IMS Connect (IMS DataStore ID).

- IXO\_DYNAMICREPLYQUEUE

MQ-CICS Bridge and MQ-IMS Bridge.

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

- IXO\_HEADERVERSION: Version of the MQCIH structure (for MQ-CICS Bridge) or of the MQIIH structure (for MQ-IMS Bridge).

1 (default) or 2, depending on the Bridge interface in use (1 character maximum).

- IXO\_LOCALCCSID: code of the character set of the machine locally MQ-CICS Bridge and MQ-IMS Bridge.  
This code (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.  
(9 characters maximum).
- IXO\_QUEUEMANAGER: Local Queue Manager name of the client part. MQ-CICS Bridge and MQ-IMS Bridge.  
(4 characters maximum).  
If there is an intermediate MQSeries Server on NT, this parameter identifies the intermediate NT Queue Manager and not the Queue Manager.
- IXO\_RACFGROUP: Name of RACF group for IMS Connect.
- IXO\_REPLYQUEUE: Name of reply messages Queue.  
MQ-CICS Bridge and MQ-IMS Bridge.  
(48 characters maximum).
- IXO\_REQUESTEXPIRY: Expiration delay of the request message, in seconds.  
MQ-CICS Bridge and MQ-IMS Bridge.  
(9 characters maximum).
- IXO\_REQUESTQUEUE: Name of request messages Queue.  
MQ-CICS Bridge and MQ-IMS Bridge.  
(48 characters maximum).
- IXO\_TIMEOUT: Maximum time required for a workstation to receive an answer from the server before indicating any communication error.  
This parameter is indicated in seconds. Its default value is 30.
- IXO\_TRANSID: Transaction code.  
(4 characters maximum for CICS and 8 characters maximum for IMS).  
This IMS transaction will have to be declared on IMS by a GEN INPUT IMS as follows:  

```
APPLCTN PSB=BVPSSOC
TRANSACTION CODE=WK35,SEGSIZE=32000,MODE=SNGL,SEGNO=00050,
          PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
          MSGTYPE=(MULTSEG,RESPONSE,$CLS)
```
- HOST\_ENCODING: Identifies the encoding of the characters used by the VisualAge Pacbase server.  
1140 (US EBCDIC) or 1146 (UK EBCDIC)  
Not used with MQ-CICS Bridge and MQ-IMS Bridge.
- MESSAGE\_LENGTH The value of this parameter MUST be 31744.

- **MONITOR:** Communication monitor code for VisualAge Pacbase, which is BVPSCPI, or BVPSSOC for MVS CICS SOCKET.

Not used with IMS Connect and MQ-IMS Bridge.

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

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

---

## Component Updating, Modification, or Removing

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

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

This option is available only with:

- Administrator & Developer workbench:  
To add one of both workbenches as the other is already installed,
- VisualAge Pacbase WorkStation:  
To add a Methodology.

- 'Remove' a component which is already installed.

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

---

## Chapter 5. Tests

---

### List of Utilities

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

Script	Description
ARCH	Archiving of the Administration Database journal
PACS	Saving of the Administration Database (SAVE)
REOR	Reorganization of the Administration Database
REST	Restoration of the Administration Database
PAGX	Extraction of Administration Database
UPGP	Update of PAF batch format of Administration Database
ARCH	Saving of the Development Database journal
PACS	Saving of the Development Database (SAVE)
PACS	Library management of the Development Database (MLIB)
REOR	Reorganization of the Development Database
REST	Restoration of the Development Database
UPDT	Batch update of the Development Database
GPRT	Generation print
PACX	Library extraction (EXLI)
PACX	Journal extraction (EXPJ)
PACX	Entities extraction (EXTR)
PACS	sub-networks extraction (UXSR)

---

### Installation Tests

The VA Pac Installation tests include the following operations:

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

## **Generation-Print, Online and Batch Update Tests**

These tests consist of the following steps:

- Online use tests:
  - Starting up the test Database server.
  - Testing screen branching.
  - Executing some updates.
- Batch updating tests:
  - Executing the UPDT procedure.
- Test on generation and print of programs:
  - Executing the 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 ARCH procedure, which outputs a PJ(1) file.
- Backup of the Administration Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file ; as the evaluation key does not allow to execute this procedure, you must enter the access key previously provided.
- Reorganization of the sequential backup, PC(1), of the Administration Database: execute the REOR procedure, which outputs a PC(2) file.
- Restoration of the Administration Database using the PJ(1) archived transaction file and the PC(2) Database backup file: execute the REST procedure.

During all these tests, the Administration Database server must be shut down.

- Extraction of Administration Database data: execute the PAGX procedure and save the GY file of extracted data.
- PAF update of the Administration Database: execute the UPGP procedure with the extracted file (result of the PAGX procedure execution).

## **Development Database Procedures Tests**

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

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



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

- Archiving of the journal created during the use tests: execute the ARCH procedure, which outputs a PJ(1) file.
- Direct backup of the Development Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file ; as the evaluation key does not allow to execute this procedure, you must enter the access key previously provided.
- Library manager: add/delete a library in the Development Database: execute the PACS (mlib) procedure, which outputs a PC(2) file.
- Reorganization of the sequential backup, PC(2), of the Development Database: execute the REOR procedure, 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 REST procedure.

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 PACX procedure (EXLI option).
- Extraction of entities from a library: execute the PACX procedure (EXTR option).
- Extraction of selected transactions and/or lists of transactions from the archived journal (PJ): execute the PACS procedure (EXPJ option).
- Extraction of sub-network: execute the PACS procedure (UXSR option).

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

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



---

## Chapter 6. Server Re-installation

---

### Re-installation

#### General Overview

You must reinstall the system environment of VisualAge Pacbase server when you receive a new version with corrections of bugs or new developments.

Generally, only system files (error messages, generation skeletons...) and programs are affected by this version.

#### GENERAL COMMENTS

- The reinstallation procedure does not create directories which are assumed to be identical to those created during the initial installation of the version.
- It does not copy the batch procedures (\$PACDIR/system/proc directory) so as not to delete possible adaptations to the site, except if the new version cannot run with the old procedures.

The new procedures are copied into the '\$PACDIR/system/proc.Vnn' (nn=version number) directory.

- Operational startup scripts, in the \$PACDIR/data/[database\_name]/script directory, are not impacted. To get the last version of the scripts, at reinstallation, refer to the startup script models in the /system/install/admin/script and /system/install/base/script directories.

**WARNING:** The metamodel of the Administration and Development Databases must be updated by executing the VINS procedure for each Database.

#### Description of the Re-Installation Procedure Steps

**WARNING:** The servers of the Administration Database and Development Databases must be stopped.

After downloading into the administrator's login directory, or mounting the CDROM, go to the following directory:

```
PB350"package_code"."version_code"
```

as in an installation operation (see sub-chapter 'VA Pac Installation'), and type the command:

```
sh pacinst.Vnn (nn = version number)
```

followed by a carriage return (Enter or Return key).

## Description of steps

The steps of the reinstallation procedure are as follows:

- consistency check on the environment variables (.profile, .kshrc or .login file),
- display of the reinstallation menu,
- choice of COBOL runtime in use (if necessary),
- distribution of the files coming from PACBASE.xxx,
- update of the journal\_pacinstall file.

## Consistency check of the environment variables

See subchapter 'Installation'.

## Display of the reinstallation menu

The reinstallation menu is displayed as follows:

```
*****
                          VisualAge Pacbase
*****
reinst      : version reinstallation
*****
Type the command name or 'x' to exit pacinst.Vnn :
```

## Description of commands

- You perform an overall reinstallation via the 'reinst' command.

Enter the command and follow the instructions given by pacinst.Vnn.

## Choice of COBOL runtime in use

See the subchapter 'Installation'.

## Update of the journal\_pacinstall file

The journal\_pacinstall file also includes information about the reinstallation:

- reinstallation date,
- release number,
- version number.

---

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

---

### Retrieval of VisualAge Pacbase 2.0 and 2.5

#### Foreword

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

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

There are two types of procedures:

#### Procedures to be executed on the 2.5 Database.

See the manual related to the 2.5 utilities used to retrieve a 2.5 version.

These procedures are :

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

#### Procedures to be executed on the new Database.

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

These procedures are :

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

#### Notes

The files which come from the retrieval must be converted, if needed, to IBM-923 code page if they are not already in ISO8859 code page.

The 'bvptrans' utility supplied in a 3.n environment enables to convert a file from one given code page to the IBM-923 code page. If needed, this utility should be applied to the PC, PE, PJ, PP, PG files which come from the 2.5 version in order to be retrieved properly in the 3.n version.

Example :

'IN' file, which comes from a 2.5 version, to convert from a PC850 code page to the 932 code page of the 3.n Repository:

```
bpstrans F-IN F-OUT 850 923
```

Note: Before the transfer of a 2.5 Database, you must execute the LVBL procedure followed by a REOR procedure on the source platform.

## **Operations to be Performed**

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

### Sequence of operations

It consists of seven steps:

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

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

2) Installation of the Administration Database

To install the Administration Database, execute the installation process.

This step creates the AN, AR, AY, AJ and GU files.

You must execute the following procedures:

- Creation of the Administration Database,
  - INGU: creation and initialization of GU user codes file,
  - REST: initialization of the Administration Database with installation data,
  - VINS: installation of the Administration Model (see the Appendix, at the end of the manual),
- Implementation of the access key from Administrator workbench by executing the following operations:
  - Input of access key,
  - targets definition,
  - key activation.

- Re-organization of the Administration Database if it is mentioned in the execution report of the preceding script (VINS).
  - PACS (SAVE): backup of the Administration Database,
  - REOR: re-organization of the Administration Database,
  - ARCH: initialization of the Administration Database journal file,
  - REST: restoration of the Administration Database.
- Retrieval of the old Database data,
  - PE25: retrieval of user parameters from the PE file which was generated during step 1,
  - UTMP: retrieval of 2.0 user passwords (optional),
  - PACS (SAVE): backup of the Administration Database.

### 3) Retrieval of a Development Database.

This step can be executed only if the corresponding VA Pac Database is already installed, with the test data.

You must execute the following procedures:

- PC25: retrieval and re-organization of the old Development Database from the backup of the old Database which was created during step 1,
- REOR: re-organization of the new Database,
- ARCH: archiving of the Development Database,
- REST: restoration of the new Development Database from the backup obtained previously; do not forget to specify the Development Database code in the user input of the procedure,
- VINS: installation of the Development Model of the new Database (see the description in the appendix at the end of this manual and create the MR file to take its transactions into account in the re-organization),

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

- PACS (SAVE option): backup of the new Development Database,
- REOR: re-organization of new Development Database,
- REST: restoration of the Development Database from the backup file resulting from the preceding re-organization procedure.

Steps 4, 5, 6 and 7 are optional.

### 4) Retrieval of generation-print commands.

It consists in executing the following procedures:

- PG20: if retrieval of 2.0 PG file,

- PG25: if retrieval of 2.5 PG file.

#### 5) Retrieval of PJ journal

It involves executing the following procedure:

- PJ25: retrieval of PJ file data.

#### 6) Retrieval of PEI files.

It involves executing the following procedure:

- PP25: retrieval of PP file,

#### 7) Retrieval of Pac/Transfer parameters (UV).

It involves executing the following procedure:

- UV25: retrieval of UV 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 into the Administration Database.

#### Notes

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



The retrieval reinitializes the license assignments.

## PE25 - Input / Processing / Results

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

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

The 'A' line has the following structure:

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

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

- the Database code: it is the logical code, which will be indicated upon the Database restoration.
- the Database name

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
42	4	cccc	Transaction code (internal area)

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

The 'C' line has the following structure:

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

## PE25 - Description of Steps

### Processing of user parameters (PE): PTU920

Code	Physical name	Type	Label
PAC7EN	Save dir.: OLDPE	Input	User parameters, old version
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7MB	User input	Input	User input
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGY	Output	User parameter transactions (length=310)
PAC7ET	User dir.: PE25ET920	Report	Error report

### Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGZ or WGY	Input	Update transactions

Code	Physical name	Type	Label
PAC7MV	Tmp dir.: WMV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	Tmp dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Work file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Work file (length=743)

Administration Database update: PACA15

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Output	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Output	Administration Database index
PAC7AY	Admin. Database - Base dir.: AY	Output	Administration Database extension
PAC7AJ	Admin. Database - Journal dir.: AJ	Output	Administration Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)

Code	Physical name	Type	Label
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of the transactions specific to a user is preceded by a banner which contains the user's code.

Return codes:

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

### PE25 - Execution Script

```
#!/bin/sh
#0(#)VA Pac xxx xxx (R) PE25 BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     RETRIEVAL OF PE FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PE25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7EN=~BVPENV PTU920 PAC7EN `dirname $PACSAVPC`/OLDPE`
BVPMSG 1044 PE 2.5 $PAC7EN
echo "-----"
echo ""
BVPPAUSE
STATUS=`pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "PE25"
    BVPMSG 1037 $1
    BVPERR
    exit 12
fi
BVPMKDIR
# -----
```

```

. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACSAVPC.ini
PAC7EN=~BVPENV PTU920 PAC7EN `dirname $PACSAVPC`~/OLDPE~
export PAC7EN
PAC7ET=~BVPENV PTU920 PAC7ET $PACUSERS/PE25ET920.txt~
export PAC7ET
PAC7GY=~BVPENV PTU920 PAC7GY $PACTMP/WGY~
export PAC7GY
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTU920"
rtspac BVPTU920
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU920"
BVPMSG 1025
BVPERR
BVPRTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=~BVPENV PAF900 PAC7GY $PACTMP/WGY~
export PAC7GY
echo " CHECKP 4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=~BVPENV PAF900 PAC7ME $PACTMP/WME~
export PAC7ME
PAC7MV=~BVPENV PAF900 PAC7MV $PACTMP/WMV~
export PAC7MV
PAC7MW=~BVPENV PAF900 PAC7MW $PACTMP/WMW~
export PAC7MW
PAC7MX=~BVPENV PAF900 PAC7MX $PACTMP/WMX~
export PAC7MX
PAC7MY=~BVPENV PAF900 PAC7MY $PACTMP/WMY~
export PAC7MY
BVPMSG 1009 "BVPFAF900"
rtspac BVPFAF900
RETURN=$?
case $RETURN in
0)
;;

```

```

*)
  BVPMSG 1012 "BVPAF900"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac

# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PE25IEA15.txt`
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PE25IFA15.txt`
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPAACA15
RETURN=$?
case $RETURN in
0)
;;
2)
  BVPMSG 1012 "BVPACA15"
  BVPMSG 1054
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
4)
  BVPMSG 1012 "BVPACA15"
  BVPMSG 1055
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
*)
  BVPMSG 1012 "BVPACA15"
  BVPMSG 1025

```

```

BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN

```

## Retrieval of the Development Database (PC25)

### PC25 - Introduction

#### Principle

This procedure (PC25) retrieves the PC file output by the backup of the old Development Database into a PC file with the new format.

#### Execution conditions

None.

#### Printed output

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

#### Result

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

### PC25 - Notes on Data Retrieval

#### Splitting up of the comment description (-G)

The comment description is split up into several descriptions.

- Comments

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

#### Caution

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

- Generation lines

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

- Generation parameters

They include the O line type (-GO).

- Error messages management

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

- Call of entities via Relations

They include the R line type (-CR).

- Specificity of the Input Aid entity

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

### **WARNING:**

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

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

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

### Data structures table type

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

### Transformation of U type manuals

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

### Extension data: User entities, Input Aids, Report Layouts

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



## PC25 - Input / Processing / Results

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

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

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

This line must be structured in this way:

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

## PC25 - Description of Steps

General processes: PTU911

Code	Physical name	Type	Label
PAC7MC	Save dir. : OLDPC	Input	Sequential image of the network (old release)
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7MB	User input	Input	User input
PAC7PB	Tmp dir. : WPC	Output	First data record (length=153)
PAC7PE	Tmp dir. : WPE	Output	User Entity Occurrence definition (2.5 release), Report layouts, and Comments (except the calls of Input Aids) (length=193)
PAC7PG	Tmp dir. : WPG	Output	Description of Input Aids and Comments including calls of Input Aids. (length=193)
PAC7PL	Tmp dir. : WPL	Output	Definition and Description of Volumes, Definition and Description of Manuals (length=193)
PAC7PZ	Tmp dir. : WPZ	Output	User Entities and description of their Occurrences (2.5 release) (length=193)
PAC7PF	Tmp dir. : WPF	Output	Other records (length=153)

Code	Physical name	Type	Label
PAC7PM	Tmp dir. : WPM	Output	Report file (length=62)
PAC7ET	User dir. : PC25ET911	Report	Report only if absence of Database code

#### Manuals and volumes processing: PTU909

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7PB	Tmp dir. : WPC	Input	First data record
PAC7PL	Tmp dir. : WPL	Input	Definition and Description of Volumes and Manuals
PAC7PI	Tmp dir. : WPI	Output	Sorted and re-formatted Volumes Definitions and Descriptions (length=153)
PAC7PM	Tmp dir. : WPM	Input/ Output	Report file

#### Comments processing: PTU92A

Code	Physical name	Type	Label
PAC7PG	Tmp dir. : WPG	Input	Description of Input Aids and of the call of Input Aids in the Comments
PAC7PM	Tmp dir. : WPM	Input/ Output	Report
PAC7PE	Tmp dir. : WPH	Output	Description of Input Aids and of the call of Input Aids in Comments (length=193)

#### Sort of Input aids : PTU92B

Code	Physical name	Type	Label
PAC7PE	Tmp dir. : WPH	Input	Description of Input Aids and Comments including calls of Input Aids
PAC7PK	Tmp dir. : WPK	Output	Description of Input Aids and Comments including calls of Input Aids (length=193)

### Sort of Input aids : PTU92C

Code	Physical name	Type	Label
PAC7PK	Tmp dir. : WPK	Input	Description of Input Aids and Comments including calls of Input Aids
PAC7KP	Tmp dir. : WKP	Output	Description of Input Aids and Comments including calls of Input Aids (length=193)
PAC7PB	Tmp dir. : WPC	Input	First data record
PAC7PD	Tmp dir. : WPD	Output	First data record (length=153)

### Report layout processing: PTU919

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7PE	Tmp dir. : WPE	Input	User Entity Occurrences Definition (2.5 rel.), Report layouts and Comments (except calls of input aids)
PAC7PB	Tmp dir. : WPD	Input	First data record
PAC7PM	Tmp dir. : WPM	Input/Output	Report file
PAC7ZP	Tmp dir. : WEP	Output	User entity Occurrences Definition (2.5 rel.), Report layouts (length=193)
PAC7PO	Tmp dir. : WPO	Output	Comments (except the call of Input Aids) (length=153)
PAC7KP	Tmp dir. : WKP	Input	Comments (including the calls of Input Aids)
PAC7PD	Tmp dir. : WPB	Output	First data record (length=153)

### Meta entities processing: PTU912

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7PZ	Tmp dir. : WPZ	Input	User Entities (2.5 release)
PAC7PB	Tmp dir. : WPC	Input	First data record

Code	Physical name	Type	Label
PAC7ZP	Tmp dir. : WZP	Output	Development Model records (Definition and Descriptions) (length=193)

#### User entities processing: PTU913

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7PX	Tmp dir. : WEP	Input	Definition of User Entity Occurrences (2.5 release) and Report Layout
PAC7PZ	Tmp dir. : WZP	Input	Definition and Description of the Development Model and Description of User Entity Occurrences (2.5 rel.)
PAC7PB	Tmp dir. : WPB	Input	First data record
PAC7ZP	Tmp dir. : WZX	Output	Long data of the Development Model, User Entities, Report layouts, and Comments (including the calls of Input Aids ) (length=193)
PAC7PD	Tmp dir. : WPR	Output	First data record (length=153)

#### Sort of long data: PTU91A

Code	Physical name	Type	Label
PAC7PZ	Tmp dir. : WZX	Input	Intermediate extension data
PAC7ZP	Tmp dir. : WXZ	Output	Sorted long data (length=193)

#### Files merging: PTU914

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

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7ZP	Tmp dir. : WXZ	Input	Sorted long data

Code	Physical name	Type	Label
PAC7PO	Tmp dir. : WPO	Input	Comments (no call of Input Aids)
PAC7PD	Tmp dir. : WPB	Input	First data record
PAC7PI	Tmp dir. : WPI	Input	Volumes Definition and Description
PAC7PF	Tmp dir. : WPF	Input	Other records
PAC7PM	Tmp dir. : WPM	Input	Report file
PAC7PC	Save dir. : PC (+1)	Output	Sequential image of the network (present release)
PAC7ET	User dir. : PC25ET914	Report	Retrieval report

### PC25 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PC25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF PC FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PC25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7MC=~BVPENV PTU911 PAC7MC `dirname $PACSAVPC`~/OLDPC~
BVPMSG 1044 PC 2.5 $PAC7MC
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7MC" ]
then
  BVPMSG 1034 $PAC7MC
  RETURN=1
  BVPRMTMP
  exit $RETURN
fi
# -----
```

```

. $PACDIR/config/$1/PAC7AE.ini
PAC7ET=~BVPENV PTU911 PAC7ET $PACUSERS/PC25ET911.txt`
export PAC7ET
PAC7MB=$PACINPUT
export PAC7MB
. $PACDIR/config/$1/PACSAVPC.ini
PAC7MC=~BVPENV PTU911 PAC7MC `dirname $PACSAVPC`/OLDPC`
export PAC7MC
PAC7PB=~BVPENV PTU911 PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PE=~BVPENV PTU911 PAC7PE $PACTMP/WPE`
export PAC7PE
PAC7PF=~BVPENV PTU911 PAC7PF $PACTMP/WPF`
export PAC7PF
PAC7PG=~BVPENV PTU911 PAC7PG $PACTMP/WPG`
export PAC7PG
PAC7PL=~BVPENV PTU911 PAC7PL $PACTMP/WPL`
export PAC7PL
PAC7PM=~BVPENV PTU911 PAC7PM $PACTMP/WPM`
export PAC7PM
PAC7PZ=~BVPENV PTU911 PAC7PZ $PACTMP/WPZ`
export PAC7PZ
BVPMSG 1009 "BVPTU911"
rtspac BVPTU911
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU911"
BVPMSG 1025
BVPERR
BVPRTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=~BVPENV PTU909 PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PI=~BVPENV PTU909 PAC7PI $PACTMP/WPI`
export PAC7PI
PAC7PL=~BVPENV PTU909 PAC7PL $PACTMP/WPL`
export PAC7PL
PAC7PM=~BVPENV PTU909 PAC7PM $PACTMP/WPM`
export PAC7PM
BVPMSG 1009 "BVPTU909"
rtspac BVPTU909
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU909"
BVPMSG 1025

```

```

BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
PAC7PG=`BVPENV PTU92A PAC7PG $PACTMP/WPG`
export PAC7PG
PAC7PE=`BVPENV PTU92A PAC7PE $PACTMP/WPH`
export PAC7PE
PAC7PM=`BVPENV PTU92A PAC7PM $PACTMP/WPM`
export PAC7PM
BVPMSG 1009 "BVPTU92A"
rtspac BVPTU92A
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU92A"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
PAC7PE=`BVPENV PTU92B PAC7PE $PACTMP/WPH`
export PAC7PE
PAC7PK=`BVPENV PTU92B PAC7PK $PACTMP/WPK`
export PAC7PK
BVPMSG 1009 "BVPTU92B"
rtspac BVPTU92B
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU92B"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
PAC7PB=`BVPENV PTU92C PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PK=`BVPENV PTU92C PAC7PK $PACTMP/WPK`
export PAC7PK
PAC7PD=`BVPENV PTU92C PAC7PD $PACTMP/WPD`
export PAC7PD
PAC7KP=`BVPENV PTU92C PAC7KP $PACTMP/WKP`
export PAC7KP
BVPMSG 1009 "BVPTU92C"

```

```

rtspac BVPTU92C
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU92C"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=~BVPENV PTU919 PAC7PB $PACTMP/WPD`
export PAC7PB
PAC7PD=~BVPENV PTU919 PAC7PD $PACTMP/WPB`
export PAC7PD
PAC7PE=~BVPENV PTU919 PAC7PE $PACTMP/WPE`
export PAC7PE
PAC7PM=~BVPENV PTU919 PAC7PM $PACTMP/WPM`
export PAC7PM
PAC7PO=~BVPENV PTU919 PAC7PO $PACTMP/WPO`
export PAC7PO
PAC7ZP=~BVPENV PTU919 PAC7ZP $PACTMP/WEP`
export PAC7ZP
PAC7KP=~BVPENV PTU919 PAC7KP $PACTMP/WKP`
export PAC7KP
BVPMSG 1009 "BVPTU919"
rtspac BVPTU919
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU919"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=~BVPENV PTU912 PAC7PB $PACTMP/WPC`
export PAC7PB
PAC7PZ=~BVPENV PTU912 PAC7PZ $PACTMP/WPZ`
export PAC7PZ
PAC7ZP=~BVPENV PTU912 PAC7ZP $PACTMP/WZP`
export PAC7ZP
BVPMSG 1009 "BVPTU912"
rtspac BVPTU912
RETURN=$?
case $RETURN in

```



```

0)
;;
*)
  BVPMSG 1012 "BVPTU912"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7PB=~BVPENV PTU913 PAC7PB $PACTMP/WPB`
export PAC7PB

PAC7PD=~BVPENV PTU913 PAC7PD $PACTMP/WPR`
export PAC7PD
PAC7PX=~BVPENV PTU913 PAC7PX $PACTMP/WEP`
export PAC7PX
PAC7PZ=~BVPENV PTU913 PAC7PZ $PACTMP/WZP`
export PAC7PZ
PAC7ZP=~BVPENV PTU913 PAC7ZP $PACTMP/WZX`
export PAC7ZP
BVPMSG 1009 "BVPTU913"
rtspac BVPTU913
RETURN=$?
case $RETURN in
0)
  ;;
*)
  BVPMSG 1012 "BVPTU913"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
esac
# -----
PAC7PZ=~BVPENV PTU91A PAC7PZ $PACTMP/WZX`
export PAC7PZ
PAC7ZP=~BVPENV PTU91A PAC7ZP $PACTMP/WXZ`
export PAC7ZP
BVPMSG 1009 "BVPTU91A"
rtspac BVPTU91A
RETURN=$?
case $RETURN in
0)
  ;;
*)
  BVPMSG 1012 "BVPTU91A"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;

```

```

esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7ET=~BVPENV PTU914 PAC7ET $PACUSERS/PC25ET914.txt`
export PAC7ET
. $PACDIR/config/$1/PACSAVPC.ini
PACSAVPC=~BVPENV PTU914 PAC7PC $PACSAVPC`
PAC7PC=$PACSAVPC.NEW
export PAC7PC
PAC7PD=~BVPENV PTU914 PAC7PD $PACTMP/WPB`
export PAC7PD
PAC7PF=~BVPENV PTU914 PAC7PF $PACTMP/WPF`
export PAC7PF
PAC7PI=~BVPENV PTU914 PAC7PI $PACTMP/WPI`
export PAC7PI
PAC7PM=~BVPENV PTU914 PAC7PM $PACTMP/WPM`
export PAC7PM
PAC7PO=~BVPENV PTU914 PAC7PO $PACTMP/WPO`
export PAC7PO
PAC7ZP=~BVPENV PTU914 PAC7ZP $PACTMP/WXZ`
export PAC7ZP
BVPMSG 1009 "BVPTU914"
rtspac BVPTU914
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU914"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPMSG 1016 "PCBACKUP.ini"
sh $PACDIR/config/$1/PCBACKUP.ini
BVPRMTMP
exit $RETURN

```

## 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 format of the new release.

It updates the Development Database with the generation-print commands and the Administration Database with the Script 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 to on-line use.

### 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 Scripts.

## **PG20 - Input / Processing / Results**

A \* line with the user code, password and the code of the Development Database for which the Script 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

Position	Length	Value	Meaning
22	4	cccc	Database code

## PG20 - Description of Steps

### Generation-print commands formatting: PTU908

Code	Physical name	Type	Label
PAC7IN	Save dir. : OLDPG	Input	Generation-print commands, old release
PAC7OU	Tmp dir.: WPG	Output	Re-formatted generation-print commands (length=150)

### Generation-print commands processing: PTU921

Code	Physical name	Type	Label
PAC7PG	Tmp dir.: WPG	Input	Generation-print commands, old release
PAC7AE	System - Skel dir. : AE	Input	Error labels
PAC7MB	User input	Input	User Entities
PAC7GY	Tmp dir. : WGY	Output	Generation-print commands transactions (length=310)
PAC7GZ	Tmp dir. : WGZ	Output	Script line transactions (length = 310)
PAC7ET	User dir. : PG20ET921	Report	Error report

### Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index

Code	Physical name	Type	Label
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGZ or WGY	Input	Update transactions
PAC7MV	Tmp dir.: WMV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	Tmp dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Work file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Work file (length=743)

Administration Database update: PACA15

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Output	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Output	Administration Database index
PAC7AY	Admin. Database - Base dir.: AY	Output	Administration Database extension
PAC7AJ	Admin. Database - Journal dir.: AJ	Output	Administration Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions

Code	Physical name	Type	Label
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of the transactions specific to a user is preceded by a banner which contains the user's code.

Return codes:

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

Transactions formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp. dir.: WGY	Input	Update transactions
PAC7MV	Tmp. dir.: WMV	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	Tmp. dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp. dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp. dir.: WMX	Output	Work file (length=743)

Code	Physical name	Type	Label
PAC7MY	Tmp dir.: WMY	Output	Work file (length=743)

#### Development Database update: PACA15

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database data
PAC7AN	Base dir.: AN	Output	Development Database index
PAC7AY	Base dir.: AY	Output	Development Database extension
PAC7AJ	Journal dir.: AJ	Output	Development Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin Database - Base dir.: AN	Input	Administration Database index
PACGGR	Admin Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin Database - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin Database - Base dir.: GU	Input	Administration Database users
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of the transactions specific to a user is preceded by a banner which contains the user's code.

Return codes:

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

## PG20 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PG20 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF PG FILE SINCE 2.0
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PG20"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7IN=~BVPENV PTU908 PAC7IN \dirname $PACSAVPC\~/OLDPG~
BVPMSG 1044 PG 2.0 $PAC7IN
echo "-----"
echo ""
BVPPAUSE
STATUS=~pactp info $1 | grep "Server Status" | cut -d: -f2~
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "PG20"
    BVPMSG 1037 $1
    BVPERR
    exit 12
fi
BVPMKDIR
if [ ! -f "$PAC7IN" ]
then
    BVPMSG 1034 $PAC7IN
    RETURN=1
    BVPRMTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PACSAVPC.ini
PAC7IN=~BVPENV PTU908 PAC7IN \dirname $PACSAVPC\~/OLDPG~
export PAC7IN
PAC7OU=~BVPENV PTU908 PAC7OU $PACTMP/WPG~
export PAC7OU
BVPMSG 1009 "BVPTU908"
rtspac BVPTU908
RETURN=?
case $RETURN in
0)
```



```

;;
*)
  BVPMSG 1012 "BVPTU908"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7ET=`BVPENV PTU921 PAC7ET $PACUSERS/PG20ET921.txt`
export PAC7ET
PAC7GY=`BVPENV PTU921 PAC7GY $PACTMP/WGY`
export PAC7GY
PAC7GZ=`BVPENV PTU921 PAC7GZ $PACTMP/WGZ`
export PAC7GZ
PAC7MB=$PACINPUT
export PAC7MB
PAC7PG=`BVPENV PTU921 PAC7PG $PACTMP/WPG`
export PAC7PG
BVPMSG 1009 "BVPTU921"
rtspac BVPTU921
RETURN=$?
case $RETURN in
0)
  ;;
*)
  BVPMSG 1012 "BVPTU921"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=`BVPENV PAF900 PAC7GY $PACTMP/WGZ`
export PAC7GY
echo " CHECKP      4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX

```

```

PAC7MY=~BVPENV PAF900 PAC7MY $PACTMP/WMY~
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AJ.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/ADMI/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PG20IEADM.txt~
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PG20IFADM.txt~
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME~
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV~
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB~
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY~
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
;;
4)
BVPMSG 1012 "BVPACA15"
BVPMSG 1055
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
;;
esac

```

```

# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=`BVPENV PAF900 PAC7GY $PACTMP/WGY`
export PAC7GY
echo " CHECKP    4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=`BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=`BVPENV PACA15 PAC7IE $PACUSERS/PG20IEA15.txt`
export PAC7IE
PAC7IF=`BVPENV PACA15 PAC7IF $PACUSERS/PG20IFA15.txt`
export PAC7IF
PAC7ME=`BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=`BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=`BVPENV PACA15 PAC7RY $PACTMP/WRY`

```

```

export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
BVPERR
BVPRMTMP
exit $RETURN
;;
4)
BVPMSG 1012 "BVPACA15"
BVPMSG 1055
BVPERR
BVPRMTMP
exit $RETURN
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## Generation-Print Commands Retrieval (PG25)

### PG25 - Introduction

#### Principle

The PG25 procedure retrieves the 2.5 PG file, which is the sequential image of the generation-print commands, into the new format.

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

#### Execution conditions

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

## Printed output

This procedure prints a report on the errors encountered.

## Note

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

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

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

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

## **PG25 - Input / Processing / Results**

A \* line with the user code, password and the code of the Development Database for which the Script 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**

### Generation-print commands processing: PTU921

Code	Physical name	Type	Label
PAC7PG	Save dir. : OLDPG	Input	Generation-print commands, old release

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error labels
PAC7MB	User input	Input	User Entities
PAC7GY	Tmp dir. : WGY	Output	Generation-print commands transactions (length=310)
PAC7GZ	Tmp dir. : WGZ	Output	Script line transactions (length = 310)
PAC7ET	User dir. : PG25ET921	Report	Error report

#### Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGZ or WGY	Input	Update transactions
PAC7MV	Tmp dir.: WMV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	Tmp dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Work file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Work file (length=743)

#### Administration Database update: PACA15

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Output	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Output	Administration Database index
PAC7AY	Admin. Database - Base dir.: AY	Output	Administration Database extension
PAC7AJ	Admin. Database - Journal dir.: AJ	Output	Administration Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of the transactions specific to a user is preceded by a banner which contains the user's code.

Return codes:

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

Transactions formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp. dir.: WGY	Input	Update transactions
PAC7MV	Tmp. dir.: WMV	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	Tmp. dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Work file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Work file (length=743)

#### Development Database update: PACA15

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database data
PAC7AN	Base dir.: AN	Output	Development Database index
PAC7AY	Base dir.: AY	Output	Development Database extension
PAC7AJ	Journal dir.: AJ	Output	Development Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin Database - Base dir.: AN	Input	Administration Database index
PACGGR	Admin Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin Database - Base dir.: AY	Input	Administration Database extension



Code	Physical name	Type	Label
PACGGU	Admin Database - Base dir.: GU	Input	Administration Database users
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of the transactions specific to a user is preceded by a banner which contains the user's code.

Return codes:

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

### PG25 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PG25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF PG FILE SINCE 2.5
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PG25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7PG=`BVPENV PTU921 PAC7PG `dirname $PACSAVPC`~/OLDPG`
BVPMSG 1044 PG 2.5 $PAC7PG
```

```

echo "-----"
echo ""
BVPPAUSE
STATUS=`pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "PG25"
    BVPMSG 1037 $1
    BVPERR
    exit 12
fi
BVPMKDIR
if [ ! -f "$PAC7PG" ]
then
    BVPMSG 1034 $PAC7PG
    RETURN=1
    BVPRMTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PACSAVPC.ini
PAC7ET=`BVPENV PTU921 PAC7ET $PACUSERS/PG25ET921.txt`
export PAC7ET
PAC7GY=`BVPENV PTU921 PAC7GY $PACTMP/WGY`
export PAC7GY
PAC7GZ=`BVPENV PTU921 PAC7GZ $PACTMP/WGZ`
export PAC7GZ
PAC7MB=$PACINPUT
export PAC7MB
PAC7PG=`BVPENV PTU921 PAC7PG `dirname $PACSAVPC`/OLDPG`
export PAC7PG
BVPMSG 1009 "BVPTU921"
rtspac BVPTU921
RETURN=$?
case $RETURN in
0)
;;
*)
    BVPMSG 1012 "BVPTU921"
    BVPMSG 1025
    BVPERR
    BVPRMTMP
    exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=`BVPENV PAF900 PAC7GY $PACTMP/WGZ`
export PAC7GY

```

```

echo " CHECKP    4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=~BVPENV PAF900 PAC7ME $PACTMP/WME~
export PAC7ME
PAC7MV=~BVPENV PAF900 PAC7MV $PACTMP/WMV~
export PAC7MV
PAC7MW=~BVPENV PAF900 PAC7MW $PACTMP/WMW~
export PAC7MW
PAC7MX=~BVPENV PAF900 PAC7MX $PACTMP/WMX~
export PAC7MX
PAC7MY=~BVPENV PAF900 PAC7MY $PACTMP/WMY~
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AJ.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/ADMI/SEMLOCK.ini
PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PG25IEADM.txt~
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PG25IFADM.txt~
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME~
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV~
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB~
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY~
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=?
case $RETURN in
0)
;;
2)
BVPMSG 1012 "BVPACA15"

```

```

BVPMSG 1054
;;
4)
BVPMSG 1012 "BVPACA15"
BVPMSG 1055
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=`BVPENV PAF900 PAC7GY $PACTMP/WGY`
export PAC7GY
echo " CHECKP      4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=`BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/SEMLOCK.ini

```

```

PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/PG25IEA15.txt`
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/PG25IFA15.txt`
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVMSG 1012 "BVPACA15"
BVMSG 1054
BVPERR
BVRMTMP
exit $RETURN
;;
4)
BVMSG 1012 "BVPACA15"
BVMSG 1055
BVPERR
BVRMTMP
exit $RETURN
;;
*)
BVMSG 1012 "BVPACA15"
BVMSG 1025
BVPERR
BVRMTMP
exit $RETURN
;;
esac
# -----
BVMSG 1010
BVRMTMP
exit $RETURN

```

## Retrieval of PJ Transactions (PJ25)

### PJ25 - Introduction

#### Principle

This procedure (PJ25) is used to change the PJ file, which is the journal file (transactions sequential file), into a new archive file in the new version format.

## Execution conditions

None.

## Printed output

This procedure generates a transaction file which indicates the number of transactions retrieved in their initial 2.n format the number of transactions converted in the new version format and the number of written transactions.

**WARNING:** The number of written transactions can be much higher than the number of read transactions. More than one transaction of the new version can be created from an old 2.n transaction, in particular for the meta entities and user entities.

Some Input Guide call transactions might be incomplete if their content is distributed on more than one transaction and if only one of these transactions is displayed in the journal. A message is displayed in this case.

## Result

This procedure generates a PJ journal file in the new release format.

## Comments

This conversion process of the journal is optional. It should be executed if required by the batch procedures (Use of Pac/Transfer).

This retrieval procedure must be used only for a conversion from a 2.0 or 2.5 version into the new version.

To retrieve some transactions, it is sometimes necessary to search for information in the new version Database. But the corresponding data may no longer exist in the new Database (example: session or library deleted). In such a case, the old transaction is retrieved with its 2.n format.

## **PJ25 - Description of Steps**

Processing of PJ transactions sequential file: PTU918

Code	Physical name	Type	Label
PAC7PJ	Save dir.: OLDPJ	Input	journal file old version
PAC7AE	System - skel. dir.: AE	Input	Error messages
PAC7AR	DBase dir.: AR	Input	Development Database data

Code	Physical name	Type	Label
PAC7AN	DBase dir.: AN	Input	Development Database index
PAC7AY	DBase dir.: AY	Input	Development Database extension data
PAC7JP	Save dir.: PJ	Output	Journal file in the new format (length=170)
PAC7ET	User dir.: PJ25ET918	Report	Report

## PJ25 - Execution Script

```

#!/bin/sh
#@(#)VA Pac xxx xxx (R) PJ25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF PJ FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PJ25"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
. $PACDIR/config/$1/PACSAVPJ.ini
PAC7PJ=`BVPENV PTU918 PAC7PJ `dirname $PACSAVPJ`~/OLDPJ`
BVPMSG 1044 PJ 2.5 $PAC7PJ
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7PJ" ]
then
  BVPMSG 1034 $PAC7PJ
  RETURN=1
  BVPRTMP
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AY.ini
PAC7ET=`BVPENV PTU918 PAC7ET $PACUSERS/PJ25ET918.txt`
export PAC7ET

```

```

. $PACDIR/config/$1/PACSAVPJ.ini
PAC7JP=~BVPENV PTU918 PAC7JP $PACSAVPJ`
export PAC7JP
PAC7PJ=~BVPENV PTU918 PAC7PJ `dirname $PACSAVPJ`~/OLDPJ`
export PAC7PJ
BVPMSG 1009 "BVPTU918"
rtspac BVPTU918
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU918"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## Retrieval of PP file (PP25)

### PP25 - Introduction

#### Principle

This procedure retrieves the 2.0 or 2.5 PP file, which is the sequential image of the Production Environment Interface, and updates the Development Database of the installed version.

#### Execution conditions

The Development Database files must be closed in the on-line mode.

#### Printed output

This procedure outputs a report which indicates the error encountered.

#### Result

The procedure generates a transaction file which contains the existing production environments, the list of the generated entities, the default environments (-GO of the Library), the list of production sessions in the new format, and updates the Development Database of the installed version.

#### Note



Any update transactions in a session or library which is not already defined in the Database will be rejected.

The PP file may contain environments with library codes or sessions to be created or purged later in the 2.0 or 2.5 Database.

When creating environments in the oldest Session (0001 or the Library initialization Session), a default Application is automatically created in the same context.

### PP25 - Input / Processing / Results

A '\*' line with a user code and a password.

If the user code is not indicated, an error message is displayed and the procedure cannot be run.

The structure of the line is presented as follows:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Option to update the environments
			in inter-library

### PP25 - Description of Steps

Management of production environment: PTU923

Code	Physical name	Type	Label
PAC7PP	Save dir. : OLDPP	Input	Back up of production environment (old release)
PAC7AE	System - Skel. dir. : AE	Input	Error message
PAC7MB	User input	Input	User input
PAC7AR	Base dir. : AR	Input	Development Database data
PAC7AN	Base dir. : AN	Input	Development Database index
PAC7AY	Base dir. : AY	Input	Development Database extension data
PAC7GY	Tmp dir. : WGY	Output	Records of production environments (length=310)

Code	Physical name	Type	Label
PAC7ET	User dir. : PP25ET923	Report	Report in case of error

Transactions formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Input	Development Database data
PAC7AN	Database dir.: AN	Input	Development Database index
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp. dir.: WGY	Input	Update transactions
PAC7MV	Tmp. dir.: WMV	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	Tmp. dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp. dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp. dir.: WMX	Output	Work file (length=743)
PAC7MY	Tmp. dir.: WMY	Output	Work file (length=743)

Development Database update: PACA15

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database data
PAC7AN	Base dir.: AN	Output	Development Database index
PAC7AY	Base dir.: AY	Output	Development Database extension
PAC7AJ	Journal dir.: AJ	Output	Development Database journal
PAC7AE	System - Skel dir.: AE	Input	Error messages
PACGGN	Admin Database - Base dir.: AN	Input	Administration Database index

Code	Physical name	Type	Label
PACGGR	Admin Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin Database - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin Database - Base dir.: GU	Input	Administration Database users
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

The list of the transactions specific to a user is preceded by a banner which contains the user's code.

Return codes:

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

### PP25 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) PP25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF PP FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "PP25"
echo "
=====
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
```

```

BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7PP=~BVPENV PTU923 PAC7PP `dirname $PACSAVPC`/OLDPP`
BVPMSG 1044 PP 2.5 $PAC7PP
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7PP" ]
then
  BVPMSG 1034 $PAC7PP
  RETURN=1
  BVPRMTMP
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACSAVPC.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
PAC7PP=~BVPENV PTU923 PAC7PP `dirname $PACSAVPC`/OLDPP`
export PAC7PP
PAC7ET=~BVPENV PTU923 PAC7ET $PACUSERS/PP25ET923.txt`
export PAC7ET
PAC7GY=~BVPENV PTU923 PAC7GY $PACTMP/WGY`
export PAC7GY
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTU923"
rtspac BVPTU923
RETURN=$?
case $RETURN in
0)
;;
*)
  BVPMSG 1012 "BVPTU923"
  BVPMSG 1025
  BVPPERR
  BVPRMTMP
  exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=~BVPENV PAF900 PAC7GY $PACTMP/WGY`

```

```

export PAC7GY
echo " CHECKP      4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`
export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=`BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
BVPERR
BVPRTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=`BVPENV PACA15 PAC7IE $PACUSERS/PP25IEA15.txt`
export PAC7IE
PAC7IF=`BVPENV PACA15 PAC7IF $PACUSERS/PP25IFA15.txt`
export PAC7IF
PAC7ME=`BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=`BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=`BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in

```

```

0)
;;
2)
BVPMSG 1012 "BVPACA15"
BVPMSG 1054
BVPERR
BVPRMTMP
exit $RETURN
;;
4)
BVPMSG 1012 "BVPACA15"
BVPMSG 1055
BVPERR
BVPRMTMP
exit $RETURN
;;
*)
BVPMSG 1012 "BVPACA15"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## Retrieval of passwords (UTMP)

### UTMP - Introduction

#### Principle

The purpose of this procedure is to retrieve the 2.n user passwords, using as input the PE file produced by the backup of users parameters (PARM procedure), and to integrate them into the Administration Database users file.

Passwords are refreshed according to the option specified on the '\*' line.

#### Execution conditions

The users file of the Administration Database must be closed to online use.

#### Printed report

In case of a wrong creation, the procedure prints an error message in display format.

#### Result

The procedure integrates 2.0 or 2.5 user passwords into the Administration Database.

## UTMP - Description of Steps

### 2.N passwords retrieval: PTURMP

Code	Physical name	Type	Label
PAC7CE	Save dir.: OLDPE	Input	Old version user parameters
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users

## UTMP - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTMP BATCH PROCEDURE
# * -----
# *          VISUALAGE PACBASE
# *
# * -----
# *          PASSWORD RETRIEVAL
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTMP"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
. $PACDIR/config/$1/PACSAVPC.ini
PAC7CE=`BVPENV PTURMP PAC7CE `dirname $PACSAVPC`~/OLDPE`
BVPMSG 1044 PE 2.5 $PAC7CE
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
. $PACDIR/config/$1/PACGGU.ini
PAC7CE=`BVPENV PTURMP PAC7CE `dirname $PACSAVPC`~/OLDPE`
export PAC7CE
BVPMSG 1009 "BVPTURMP"
rtspac BVPTURMP
RETURN=$?
case $RETURN in
0)
;;
*)
```

```

BVPMSG 1012 "BVPTURMP"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## 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

### Processing of transfer parameters: PTU922

Code	Physical name	Type	Label
PAC7UV	Base dir.: OLDUV	Input	Transfer parameters, old release
PAC7AE	System - Skel. dir. : AE	Input	Error messages
PAC7MB	User input	Input	User input
PAC7GY	Tmp dir.: WGY	Output	Transfer parameter transactions (length=310)
PAC7ET	User dir.: UV25ET922	Report	Report in case of error

### Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGZ or WGY	Input	Update transactions
PAC7MV	Tmp dir.: WMV	Output	Formatted transactions (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	Tmp dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Work file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Work file (length=743)

## Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Output	Administration Database Data file
PAC7AN	Admin. Database - Base dir.: AN	Output	Administration Database Index file
PAC7AY	Admin. Database - Base dir.: AY	Output	Administration Database extension
PAC7AJ	Admin. Database - Base dir.: AJ	Output	Administration Database journal
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database Index file
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database Data file
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7DC	Database dir.: DC	Input	DSMS file of Development Database elements
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	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
- 4 : error

## UV25 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UV25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF UV FILE
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UV25"
echo "      ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
. $PACDIR/config/$1/PAC7AR.ini
PAC7UV=`BVPENV PTU922 PAC7UV `dirname $PAC7AR`/OLDUV`
BVPMSG 1044 UV 2.5 $PAC7UV
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7UV" ]
then
    BVPMSG 1034 $PAC7UV
    RETURN=1
    BVPRTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
PAC7UV=`BVPENV PTU922 PAC7UV `dirname $PAC7AR`/OLDUV`
export PAC7UV
PAC7ET=`BVPENV PTU922 PAC7ET $PACUSERS/UV25ET922.txt`
export PAC7ET
PAC7GY=`BVPENV PTU922 PAC7GY $PACTMP/WGY`
export PAC7GY
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTU922"
rtspac BVPTU922
RETURN=?
case $RETURN in
0)
;;
*)
    BVPMSG 1012 "BVPTU922"
    BVPMSG 1025
```

```

BVPERR
BVPRMTP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7GY=`BVPENV PAF900 PAC7GY $PACTMP/WGY`
export PAC7GY
echo " CHECKP    4000UPDT" > $PAC7GY.tmp
cat $PAC7GY >> $PAC7GY.tmp
mv $PAC7GY.tmp $PAC7GY
PAC7ME=`BVPENV PAF900 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=`BVPENV PAF900 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7MW=`BVPENV PAF900 PAC7MW $PACTMP/WMW`

export PAC7MW
PAC7MX=`BVPENV PAF900 PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7MY=`BVPENV PAF900 PAC7MY $PACTMP/WMY`
export PAC7MY
BVPMSG 1009 "BVPAF900"
rtspac BVPAF900
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPAF900"
BVPMSG 1025
BVPERR
BVPRMTP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/ADMI/PAC7AJ.ini
. $PACDIR/config/ADMI/PAC7AN.ini
. $PACDIR/config/ADMI/PAC7AR.ini
. $PACDIR/config/ADMI/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini

```

```

PAC7IE=~BVPENV PACA15 PAC7IE $PACUSERS/UV25IEA15.txt`
export PAC7IE
PAC7IF=~BVPENV PACA15 PAC7IF $PACUSERS/UV25IFA15.txt`
export PAC7IF
PAC7ME=~BVPENV PACA15 PAC7ME $PACTMP/WME`
export PAC7ME
PAC7MV=~BVPENV PACA15 PAC7MV $PACTMP/WMV`
export PAC7MV
PAC7RB=~BVPENV PACA15 PAC7RB $PACTMP/WRB`
export PAC7RB
PAC7RY=~BVPENV PACA15 PAC7RY $PACTMP/WRY`
export PAC7RY
BVPMMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
;;
2)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1054
BVPERR
BVPRTMP
exit $RETURN
;;
4)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1055
BVPERR
BVPRTMP
exit $RETURN
;;
*)
BVPMMSG 1012 "BVPACA15"
BVPMMSG 1025
BVPERR
BVPRTMP
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
BVPRTMP
exit $RETURN

```

## MB Transactions exchanges between 2.n & 3.n (MB25)

### MB25 - Introduction

#### Principle

This procedure retrieves the 2.0 or 2.5 UPDT format transactions in the new release.

## Limits

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

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

The P.I.A. types change between the 2.n and 3.n versions: to allow the exchange of transactions between these 2 versions, execute, in the 2.n version, the UTFG procedure which assigns the types in the 3.n version.

## Execution conditions

None.

## Printed output

This procedure prints a report of the errors encountered.

## Result

This procedure generates a transaction file for the new version UPDT procedure and a revoked transactions file.

## **MB25 - Description of Steps**

### MB file retrieval: PTU926

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7AR	Database dir.: AR	Input	Development Database data
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7MB	User dir.: OLDMB	Input	2.5 MB transactions
PAC7MV	User dir.: MB25MV	Output	Retrieval transactions for UPDT
PAC7ME	User dir.: MB25ME	Output	Revoked transactions
PAC7EF	User dir.: MB25EF926	Report	Retrieval reports
PAC7DD	User dir.: MB25DD926	Report	Batch procedure authorization option

## MB25 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) MB25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF MB FILE
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "MB25"
echo "      ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7MB=`BVPENV PTU926 PAC7MB `dirname $PACUSERS`/OLDMB`
export PAC7MB
BVPMSG 1044 MB 2.5 $PAC7MB
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7MB" ]
then
  BVPMSG 1034 $PAC7MB
  RETURN=1
  BVPRTMP
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=`BVPENV PTU926 PAC7DD $PACUSERS/MB25DD926.txt`
export PAC7DD
PAC7EF=`BVPENV PTU926 PAC7EF $PACUSERS/MB25EF926.txt`
export PAC7EF
PAC7MB=`BVPENV PTU926 PAC7MB `dirname $PACUSERS`/OLDMB`
export PAC7MB
PAC7ME=`BVPENV PTU926 PAC7ME $PACUSERS/MB25ME`
export PAC7ME
PAC7MV=`BVPENV PTU926 PAC7MV $PACUSERS/MB25MV`
export PAC7MV
BVPMSG 1009 "BVPTU926"
rtspac BVPTU926
```

```

RETURN=$?
case $RETURN in
0)
  ;;
*)
  BVPMSG 1012 "BVPTU926"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## GY Transactions exchanges between 2.n & 3.n (GY25)

### GY25 - Introduction

#### Principle

This procedure retrieves the 2.0 or 2.5 UPDP format transactions into the new release.

#### Limits

The User Entities, Relations, U-type Manuals are not processed.

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

The P.I.A. types change between the 2.n and 3.n versions: to allow the transactions exchange between these versions, execute in 2.n the UTFG procedure which assigns the 3.n types.

#### Execution conditions

None.

#### Printed output

This procedure prints a report of the errors encountered.

#### Result

This procedure generates a transaction file for the UPDP procedure of the new release and a revoked transactions file.



## GY25 - Description of Steps

GY file retrieval: PTU927

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7AR	Database dir.: AR	Input	Development Database data
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7GY	User dir.: OLDGY	Input	2.5 GY transactions
PAC7MV	User dir.: GY25MV	Output	Retrieval transactions for UPDP
PAC7ME	User dir.: GY25ME	Output	Revoked transactions
PAC7EF	User dir.: GY25EF927	Report	Retrieval reports
PAC7DD	User dir.: GY25DD927	Report	Authorization option

## GY25 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) GY25 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      RETRIEVAL OF GY FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "GY25"
echo "      ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7GY=`BVPENV PTU927 PAC7GY `dirname $PACUSERS`~/OLDGY`
export PAC7GY
BVPMSG 1044 GY 2.5 $PAC7GY
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
```

```

if [ ! -f "$PAC7GY" ]
then
  BVPMSG 1034 $PAC7GY
  RETURN=1
  BVPRMTMP
  exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=~BVPENV PTU927 PAC7DD $PACUSERS/GY25DD927.txt`
export PAC7DD
PAC7EF=~BVPENV PTU927 PAC7EF $PACUSERS/GY25EF927.txt`
export PAC7EF
PAC7GY=~BVPENV PTU927 PAC7GY `dirname $PACUSERS\`/OLDGY`
export PAC7GY
PAC7ME=~BVPENV PTU927 PAC7ME $PACUSERS/GY25ME`
export PAC7ME
PAC7MV=~BVPENV PTU927 PAC7MV $PACUSERS/GY25MV`
export PAC7MV
BVPMSG 1009 "BVPTU927"
rtspac BVPTU927
RETURN=$?
case $RETURN in
0)
  ;;
*)
  BVPMSG 1012 "BVPTU927"
  BVPMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## MB Transactions exchanges between 3.n & 2.n (MB30)

### MB30 - Introduction

#### Principle

This procedure retrieves the UPDT format transactions of the new release to the 2.0 or 2.5 release.

#### Limits

The User Entities, Meta entities, Relations, Descriptions, Keywords, associated Documentation and Relation calls are not processed.

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

This procedure generates a transaction file for the 2.0 or 2.5 UPDT procedure and a revoked transactions file.

**MB30 - Description of Steps**

MB file retrieval: PTU928

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AR	Database dir. : AR	Input	Development Database data
PACGGR	Admin Database - Base dir. : AR	Input	Administration Database data
PACGGN	Admin Database - Base dir. : AN	Input	Administration Database index
PACGGU	Admin Database - Base dir. : GU	Input	Administration Database users
PAC7MB	User dir.: OLDMB	Input	MB transactions of the new version
PAC7MV	User dir. : MB30MV	Output	Retrieval transactions for UPDT
PAC7ME	User dir. : MB30ME	Output	Revoked transactions
PAC7EF	User dir. : MB30EF928	Report	Retrieval reports
PAC7DD	User dir. : MB30DD928	Report	Authorization option

**MB30 - Execution Script**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) MB30 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
```

```

# *                RETRIEVAL OF MB FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "MB30"
echo "                ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7MB=`BVPENV PTU928 PAC7MB \dirname $PACUSERS\OLDDB`
export PAC7MB
BVPMSG 1044 MB 3.5 $PAC7MB
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7MB" ]
then
    BVPMSG 1034 $PAC7MB
    RETURN=1
    BVPRMTMP
    exit $RETURN
fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=`BVPENV PTU928 PAC7DD $PACUSERS/MB30DD928.txt`
export PAC7DD
PAC7EF=`BVPENV PTU928 PAC7EF $PACUSERS/MB30EF928.txt`
export PAC7EF
PAC7MB=`BVPENV PTU928 PAC7MB \dirname $PACUSERS\OLDDB`
export PAC7MB
PAC7ME=`BVPENV PTU928 PAC7ME $PACUSERS/MB30ME`
export PAC7ME
PAC7MV=`BVPENV PTU928 PAC7MV $PACUSERS/MB30MV`
export PAC7MV
BVPMSG 1009 "BVPTU928"
rtspac BVPTU928
RETURN=$?
case $RETURN in
0)
;;
*)
    BVPMSG 1012 "BVPTU928"
    BVPMSG 1025
    BVPERR
    BVPRMTMP

```

```

    exit $RETURN
    ;;
  esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## GY Transactions exchanges between 3.n & 2.n (GY30)

### GY30 - Introduction

#### Principle

This procedure retrieves the UPDP format transactions of the new release into the 2.0 or 2.5 release.

#### Limits

The Meta entities, Relations, Descriptions, Keywords, associated Documentation and Relation calls are not processed.

But, User entities are retrieved as 2.n User entity occurrences. The user will have to create the corresponding 2.n user entity.

#### Execution conditions

None.

#### Printed output

This procedure prints a report of the errors encountered.

#### Result

This procedure generates a transaction file for the 2.0 or 2.5 UPDP procedure and a revoked transactions file.

### GY30 - Description of Steps

#### GY file retrieval: PTU929

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AR	Database dir. : AR	Input	Development Database data
PAC7AN	Database dir. : AN	Input	Development Database index

Code	Physical name	Type	Label
PACGGR	Admin Database - Base dir. : AR	Input	Administration Database data
PACGGN	Admin Database - Base dir. : AN	Input	Administration Database index
PACGGU	Admin Database - Base dir. : GU	Input	Administration Database users
PAC7GY	User dir. : OLDGY	Input	GY transactions of the new version
PAC7MV	User dir. : GY30MV	Output	Retrieval transactions for UPDP
PAC7ME	User dir. : GY30ME	Output	Revoked transactions
PAC7EF	User dir. : GY30EF929	Report	Retrieval reports
PAC7DD	User dir. : GY30DD929	Report	Authorization option

### GY30 - Execution Script

```

#!/bin/sh
#@(#)VA Pac xxx xxx (R) GY30 BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *             RETRIEVAL OF GY FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "GY30"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
PAC7GY=`BVPENV PTU929 PAC7GY \dirname $PACUSERS\~/OLDGY`
export PAC7GY
BVPMSG 1044 GY 3.5 $PAC7GY
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
if [ ! -f "$PAC7GY" ]
then
    BVPMSG 1034 $PAC7GY
    RETURN=1
    BVPRMTP
    exit $RETURN

```

```

fi
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
PAC7DD=`BVPENV PTU929 PAC7DD $PACUSERS/GY30DD929.txt`
export PAC7DD
PAC7EF=`BVPENV PTU929 PAC7EF $PACUSERS/GY30EF929.txt`
export PAC7EF
PAC7GY=`BVPENV PTU929 PAC7GY `dirname $PACUSERS`/OLDGY`
export PAC7GY
PAC7ME=`BVPENV PTU929 PAC7ME $PACUSERS/GY30ME`
export PAC7ME
PAC7MV=`BVPENV PTU929 PAC7MV $PACUSERS/GY30MV`
export PAC7MV
BVPMSG 1009 "BVPTU929"
rtspac BVPTU929
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTU929"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## Procedures - Summary Table of Changes

List of new procedures since the 2.5

Procedure	Comments
ARCH	Archiving Administration Database
ARPM	Archiving the QJ journal of SCM module
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
INQJ	Initialization of SCM Tools Interface archive journal
GPMC	Generation of MOVE CORRESPONDING
GPRC	Generation (COBOL API)
REOR	Re-organization of Administration Database

<b>Procedure</b>	<b>Comments</b>
REST	Restoration of Administration Database
PACS	Backup of Administration Database (SAVE option)
PAGX	Extraction of Administration Database
UKD1	Batch update of access keys and rights
VINS	Update of Administration Model
PACS	Management of Development Database
GY25	Retrieval of GY file for UPDP to 3.n
MB25	Retrieval of MB file for UPDT to 3.n
GY30	Retrieval of GY file for UPDP to 2.n
MB30	Retrieval of MB file for UPDT to 2.n
PC25	Retrieval of Development Database
PE25	Retrieval of user parameters (PE)
PG20	Retrieval of 2.0 generation-print commands (PG)
PG25	Retrieval of 2.5 generation-print commands (PG)
PJ25	Retrieval of 2.n journal
PP25	Retrieval of PP file
RPP2	Retrieval of PEI environment
UTMP	Retrieval of passwords
UV25	Retrieval of PacTransfer parameters (UV)
UTM1	Conversion of meta entities (step 1)
UTM2	Conversion of meta entities (step 2)
UTU1	Extraction of 'P' lines with 'UNS'
UTU2	Update of 'P' lines with 'UNS'
STAT	Database statistics
VINS	Update of Development Model
CHPM	Integrity control of environments/elements
CPPM	Comparison between Database and user configuration
EXPM	Extraction of environments
GPPM	Generation report written into QJ
HIPM	Database automatic freeze
PUPM	Purge of SCM entities
SIPM	Generation simulation
UPPM	Update of elements



Procedure	Comments
TRED	PacTransfer: print of parameters
UPGP	PAF update of Administration Database

List of procedures suppressed since 2.5 release

Procedure	Program	Comments
CPSN	PTU850 PTU855	Integrated into PACX procedure
EMSN	PTU810	
MESN	PTU815	
SASN	PTU130 PTU140	Integrated into PACS procedure
UXSR	UTIXSR	Integrated into PACS procedure
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACT11	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	
RVKE	PREI40 PREI50	
STOP	PTUR00	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	
VPU2	PVA320	

Procedure	Program	Comments
LVBL	PTULVB	
QREO	PTUN00 PTUN10 PTUN40	
RPPG	PTU908	
RPTD	PTAR20	

---

## Retrieval of VisualAge Pacbase 3.0

### Operations to be Performed

The 3.5 version can be installed in the same environment as the 3.0 version. This case is considered as a re-installation.

Before, it is recommended to save the Development and Administration Databases (SAVE, ARCH) in 3.0 version.

- If this version is installed in the same environment as the 3.0, refer to the 'Re-installation of Server' chapter in this manual.  
After the Administration Model update (VINS execution), connect to the Administrator workbench to enter the new access key.
- If this version is installed in a new environment, after the execution of all the steps of the server part installation, the Databases must be restored following the steps:
  - REST : restoration of the Administration Database,
  - VINS : update of the Administration Model,
  - Connection via the Administration workbench to enter and activate the new access key to the system,
  - REST : restoration of each Development Database,
  - VINS : update of the Development Model for each Development Database.
- In any case, the reorganization of Databases is required (ARCH, SAVE, REOR, REST).

---

## Chapter 8. Components

---

### Server Environment Components

#### Introduction

One of the purposes of the product is to manage permanent data in either batch or on-line mode, by using two types of resources:

- Directories in which the programs which make up the system, and the parameters required to run the system, are stored.
- 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.

#### On-Line Documentation

Besides the libraries described in the preceding subchapters, the VA PAC system includes the AE file which contains the error messages and on-line documentation.

#### Generation Skeletons

The product also includes the following files:

- A generation skeleton file (SA file) used by the Batch generator function for COBOL API.
- A generation skeleton file (SC file) used by the Batch generator function.
- A generation skeleton file (SG file) used by the On-Line Systems Development and Database generator functions.
- A generation skeleton file (SN file) used by the eBusiness generator function.
- A generation skeleton file (SR file), used by the Reverse generator function.
- A generation skeleton file (SP file) used by the PAF function for the generation of extractors.
- A skeleton file (SF file) used by the PAF function for the generation of extractors.
- A generation skeleton file (SS file), used by the eBusiness generator function.

## Administration Database

### Administration Database Files

- Data file, AR.
- Extension data file, AY.
- Index file, AN.
- Journal file, AJ.
- User file, GU.

This file is sensitive and its backup must be executed in a secure environment of the installation site.

- Journal file of SCM Tools Interface, QJ

### Administration Database Backup

The Administration Database backup consists of two sequential generation files.

- Backup of the Database (PC).

This is a backup file of the Administration Database components: index (AN), Data (AR) and extension (AY) in a sequential format.

- Backup of the journal (PJ).

The purpose of this file is to store all the update transactions that have affected the Administration Database since its installation and that have passed through the transactions file (AJ).

When the size of this file becomes incompatible with operation requirements, the ARCH procedure of the Administration Database enables you to split it into several files, among which only the most recent one is used on a regular basis.

- Backup of SCM module QJ journal

The backup of QJ is the JQ file.

The purpose of this file is to archive the valid transactions already processed by different updates and stored in the QJ journal file.

## Development Database

### Development Database Files

These files contain all the data related to development of applications.

- The Data file (AR).

- The extension data file (AY).
- The index file (AN).
- The journal file (AJ).

All the transactions performed on the Database in batch or on-line mode are saved for two reasons:

- To allow Database restoration if the system standard securities were to fail.
- This information may be used for statistical purposes.

These transactions are usually stored in the transactions backup file (PJ). The transactions file is used temporarily, between the moment transactions are processed by the system and the moment they are saved on their final storage medium by the ARCH procedure.

### **Development Database Backup Files**

According to the organization taken into account upon restoration, the Database backup is either made of two sequential 'rotating' files (PC and PJ) or of four sequential 'rotating' 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)).

- The backup of the Development Database index (PD).
- The backup of the Development Database bulk data (PY).
- 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.

### **Modules - Specific Files**

#### **Pac/Impact:**

- File of already-impacted criteria (FQ).
- Search criteria or entry points file (FH).
- Reduced file of criteria for purge (FR).
- Impact result file (FO).
- File of entities to be analyzed (FP).

## DSMS:

When the DSMS function (refer to the 'DSMS' manual) is available on site, a DSMS file is accessed for each Development Database, in batch and on-line modes.

This file contains the list of the entities concerned by each change. The change number is entered by the user on the Database sign-on screen.

- The DSMS file of the Development Database elements (DC)

This file is allocated and initialized at the time of the installation of the DSMS Function.

The definition supplied when installing VA Pac must be used if the DSMS has not been installed on the site yet.

## PAF:

- PAF work file (PA) for PAF-TP and PUF-TP.

All the user on-line programs which access Databases with the same root need an indexed work file to use the PAF and PUF functions.

- Work file for PAF in batch mode

All the user batch programs need an indexed work file to use the PAF function. This file is allocated for the job duration and is destroyed at the end of the job.

## PAF extension

Extraction master path file (GS), containing the user's extractors and macro-commands.

## Complementary Libraries and Files

Complementary Dictionary files are located in the \SYS\SKEL.

## PQC function

- The BVPQCRA and BVPQCRF files contain the standard quality rules.

- Specific members of Pacbench Quality Control function

File	Contents of format	Comments
BVPQCRA	Sequential file in English	Standard rules
BVPQCRF	Sequential file in French	Standard rules

MIAM procedure.

The parameters file for the MIAM procedure is automatically supplied in the \SYS\SKEL directory.

<b>Member</b>	<b>Content or format</b>	<b>Remarks</b>
BVPAMIAM	Sequential file	Standard parameters for the procedure MIAM





---

## Chapter 9. Appendix

---

### Installation of the Administration Database Model

#### VINS - Introduction

The VINS procedure performs the batch update of the Administration Database using transactions provided by IBM. Another file may sometimes be used. For example when the Pacdesign/Pacbench methodology changes.

#### Execution conditions

The Database must be closed to on-line processing.

#### Abnormal execution

Refer to the Administrator's Procedures manual, sub-chapter 'Abnormal Ending'.

When an abend occurs during the execution of the BVPACI30 or BVPACI40 program, the Database is no longer consistent.

Once the problem has been solved, the Database must be reloaded with a retrieval of archived transactions and the VINS procedure must be executed again.

#### VINS - Input / Processing / Results

This procedure requires two types of user input:

- A line which contains the User ID as well as the operation to perform,
- The transactions which enable the creation of IBM Meta Entities and the retrieval of client User Entities with the 'extension' format: the user should never modify the content of these transactions.

The structure of the line is the following:

Position	Length	Value	Meaning
2	1	'*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Library code
29	4	'VINS'	

Position	Length	Value	Meaning
33	1	T	Installation of IBM Meta Entities

### Printed output

The procedure outputs:

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

### Result

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

### Note

The transactions extracted for the REOR procedure are stored in a file, assigned by PAC7MR, which is to be made permanent by substitution. otherwise):

## **VINS - Description of Steps**

### Update of the Administration Database: VINS

Code	Physical name	Type	Label
PAC7AE	System - skel. dir.: AE	Input	Error labels
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database Index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database Data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database Users
PAC7AJ	Journal directory: AJ	Output	Administration Database Journal
PAC7AN	Admin. Database - Base dir.: AN	Output	Administration Database Index
PAC7AR	Admin. Database - Base dir.: AR	Output	Administration Database Data

Code	Physical name	Type	Label
PAC7AY	Admin. Database - Base dir.: AY	Output	Administration Database Extension
PAC7MA	System - skel. dir.: BVPMETAD	Input	IBM Meta Entities Transactions
PAC7MB	User input	Input	User Input
PAC7BM	Tmp dir.: WBM	Input/ Output	Work file
PAC7WD	Tmp dir.: WWD	Input/ Output	Extracted Transactions
PAC7ES	Tmp dir.: WES	Input/ Output	Extracted Transactions
PAC7TD	User dir.: TD	Input/ Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR on Administration Base
PAC7MX	Tmp dir.: WMX	Output	Non extracted entities
PAC7RQ	Tmp dir.: WRQ	Output	Work file
PAC7IA	User dir.: VINSIA	Report	Complete printing of programs sequence
PAC7CP	/dev/null		
PAC7EE	User dir.: VINSEE	Report	Report
PAC7EQ	User dir.: VINSEQ	Report	Report
PAC7EU	User dir.: VINSEU	Report	Report
PAC7ER	User dir.: VINSER	Report	Report
PAC7EZ	User dir.: VINSEZ	Report	Report
PAC7DD	User dir.: VINSDD	Report	Report
PAC7IE	User dir.: VINSIE	Report	Report
PAC7IF	User dir.: VINSIF	Report	Report
PAC7IG	User dir.: VINSIG	Report	Report
PAC7IH	User dir.: VINSIH	Report	Report

Return codes:

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

## VINS - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) VINS BATCH PROCEDURE
# * -----
# *     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"
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "VINS"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
. $PACDIR/config/$1/PAC7AJ.ini
BVPMSG 1015 "`dirname $PAC7AJ`"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
STATUS=`pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "VINS"
    BVPMSG 1037 $1
    BVPPER
    exit 12
fi
BVPMKDIR
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
```

```

. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
PAC7BM=`BVPENV VINS PAC7BM $PACTMP/WBM`
export PAC7BM
PAC7CP=`BVPENV VINS PAC7CP /dev/null`
export PAC7CP
PAC7DD=`BVPENV VINS PAC7DD $PACUSERS/VINSDD.txt`
export PAC7DD
PAC7EE=`BVPENV VINS PAC7EE $PACUSERS/VINSEE.txt`
export PAC7EE
PAC7EQ=`BVPENV VINS PAC7EQ $PACUSERS/VINSEQ.txt`
export PAC7EQ
PAC7ER=`BVPENV VINS PAC7ER $PACUSERS/VINSER.txt`
export PAC7ER
PAC7ES=`BVPENV VINS PAC7ES $PACTMP/WES`
export PAC7ES
PAC7EU=`BVPENV VINS PAC7EU $PACUSERS/VINSEU.txt`
export PAC7EU
PAC7EZ=`BVPENV VINS PAC7EZ $PACUSERS/VINSEZ.txt`
export PAC7EZ
PAC7IA=`BVPENV VINS PAC7IA $PACUSERS/VINSIA.txt`
export PAC7IA
PAC7IE=`BVPENV VINS PAC7IE $PACUSERS/VINSIE.txt`
export PAC7IE
PAC7IF=`BVPENV VINS PAC7IF $PACUSERS/VINSIF.txt`
export PAC7IF
PAC7IG=`BVPENV VINS PAC7IG $PACUSERS/VINSIG.txt`
export PAC7IG
PAC7IH=`BVPENV VINS PAC7IH $PACUSERS/VINSIH.txt`
export PAC7IH
. $PACDIR/config/$1/PACVINS.ini
PAC7MA=`BVPENV VINS PAC7MA $PACVINS`
export PAC7MA
PAC7MB=$PACINPUT
export PAC7MB
PAC7MR=`BVPENV VINS PAC7MR $PACTMP/WMR`
export PAC7MR
PAC7MX=`BVPENV VINS PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7RQ=`BVPENV VINS PAC7RQ /dev/null`
export PAC7RQ
PAC7TD=`BVPENV VINS PAC7TD $PACUSERS/TD.txt`
export PAC7TD
PAC7WD=`BVPENV VINS PAC7WD $PACTMP/WWD`
export PAC7WD
SYSEXT=`BVPENV VINS SYSEXT $PACTMP/WSY`
export SYSEXT
BVPMSG 1009 "BVPVINS"
rtspac BVPVINS
RETURN=$?
case $RETURN in

```

```

0)
;;
4)
  BVPMMSG 1008 $PACUSERS
  ;;
*)
  BVPMMSG 1012 "BVPVINS"
  BVPMMSG 1025
  BVPERR
  BVPRMTMP
  exit $RETURN
  ;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN

```

---

## Installation of the Development Database Model

### VINS - Introduction

The VINS procedure performs the batch update of the Development Database using transactions provided by IBM. Another file may sometimes be used (for example when the Pacdesign/Pacbench methodology changes).

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

#### Execution conditions

The Database must be closed to on-line processing.

#### Abnormal execution

Refer to the 'Administrator's Procedures' manual, sub-chapter 'Abnormal Ending'.

When an abend occurs during the execution of the BVPACI30 or BVPACI40 programs, the Database is no longer consistent.

Once the problem has been solved, the Database must be reloaded with a retrieval of archived transactions and the VINS procedure must be executed again.

### VINS - Input / Processing / Results

This procedure requires two types of user input:

- a line which contains the User ID as well as the operation to perform,

- the transactions which enable the creation of IBM Meta Entities and the retrieval of client User Entities with the 'extension' format: the user should never modify the content of these transactions.

The structure of the line is the following:

Position	Length	Value	Meaning
2	1	'*	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Library code
29	4	'VINS'	
33	1	'I'	Installation of IBM Meta Entities
		'R'	Retrieval of User Entities with the 'extension' format
		' '	'I' + 'R'

### Printed output

The procedure prints

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

### Result

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

For the retrieval of User Entities with an extension format, a sequential file of purge transactions can be generated (if 'R' in column 33 in the user input). You then have to reorganize the Database with this file as input. The reorganization of the Database with the PC file, saved after the retrieval, is also required if the client user entities of the .PPTX and .QPAQC meta entities exist in more than one session.

### Note

The transactions extracted for the REOR procedure are stored in a file, assigned by PAC7MR, which is to be made permanent by substitution.

## VINS - Description of Steps

### Update of the Development Database : VINS

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error Labels
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database Index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database Data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database Users
PAC7AJ	Journal dir.: AJ	Output	Development Database Journal
PAC7AN	Base dir.: AN	Output	Development Database Index
PAC7AR	Base dir.: AR	Output	Development Database Data
PAC7AY	Base dir.: AY	Output	Development Database Extension
PAC7MA	System - skel. dir.: BVPMETBA	Input	IBM Meta Entities Transactions
PAC7MB	User input	Input	User Input
PAC7BM	Tmp dir.: WBM	Input/ Output	User input
PAC7WD	Tmp dir.: WWD	Input/ Output	Extracted Transactions
PAC7ES	Tmp dir.: WES	Input/ Output	Extracted Transactions
PAC7TD	User dir.: TD	Input/ Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR
PAC7MX	Tmp dir.: WMX	Output	Non extracted entities
PAC7RQ	Tmp dir.: WRQ	Output	Work file
PAC7IA	User dir.: VINSIA	Report	Complete printing of programs sequence
PAC7CP	/dev/null		
PAC7EE	User dir.: VINSEE	Report	Report
PAC7EQ	User dir.: VINSEQ	Report	Report
PAC7EU	User dir.: VINSEU	Report	Report



Code	Physical name	Type	Label
PAC7ER	User dir.: VINSER	Report	Report
PAC7EZ	User dir.: VINSEZ	Report	Report
PAC7DD	User dir.: VINSDD	Report	Report
PAC7IE	User dir.: VINSIE	Report	Report
PAC7IF	User dir.: VINSIF	Report	Report
PAC7IG	User dir.: VINSIG	Report	Report
PAC7IH	User dir.: VINSIH	Report	Report

Return codes:

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

## VINS - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) VINS BATCH PROCEDURE
# * -----
# *          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"
# * -----
# *
# * Parameter control
# * . $PACDIR/system/proc/BVPINIT.ini
# * echo ""
# * echo "-----"
# * BVPMSG 1004 "VINS"
```

```

echo "                               ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
. $PACDIR/config/$1/PAC7AJ.ini
BVPMSG 1015 "`dirname $PAC7AJ`"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
STATUS=`pactp info $1 | grep "Server Status" | cut -d: -f2`
if [ "$STATUS" != "Inactive" -a "$STATUS" != "" ]
then
    BVPMSG 1012 "VINS"
    BVPMSG 1037 $1
    BVPERR
    exit 12
fi
BVPMKDIR
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
PAC7BM=`BVPENV VINS PAC7BM $PACTMP/WBM`
export PAC7BM
PAC7CP=`BVPENV VINS PAC7CP /dev/null`
export PAC7CP
PAC7DD=`BVPENV VINS PAC7DD $PACUSERS/VINSDD.txt`
export PAC7DD
PAC7EE=`BVPENV VINS PAC7EE $PACUSERS/VINSEE.txt`
export PAC7EE
PAC7EQ=`BVPENV VINS PAC7EQ $PACUSERS/VINSEQ.txt`
export PAC7EQ
PAC7ER=`BVPENV VINS PAC7ER $PACUSERS/VINSER.txt`
export PAC7ER
PAC7ES=`BVPENV VINS PAC7ES $PACTMP/WES`
export PAC7ES
PAC7EU=`BVPENV VINS PAC7EU $PACUSERS/VINSEU.txt`
export PAC7EU
PAC7EZ=`BVPENV VINS PAC7EZ $PACUSERS/VINSEZ.txt`
export PAC7EZ
PAC7IA=`BVPENV VINS PAC7IA $PACUSERS/VINSIA.txt`
export PAC7IA
PAC7IE=`BVPENV VINS PAC7IE $PACUSERS/VINSIE.txt`
export PAC7IE
PAC7IF=`BVPENV VINS PAC7IF $PACUSERS/VINSIF.txt`
export PAC7IF
PAC7IG=`BVPENV VINS PAC7IG $PACUSERS/VINSIG.txt`

```

```

export PAC7IG
PAC7IH=`BVPENV VINS PAC7IH $PACUSERS/VINSIH.txt`
export PAC7IH
. $PACDIR/config/$1/PACVINS.ini
PAC7MA=`BVPENV VINS PAC7MA $PACVINS`
export PAC7MA
PAC7MB=$PACINPUT
export PAC7MB
PAC7MR=`BVPENV VINS PAC7MR $PACTMP/WMR`
export PAC7MR
PAC7MX=`BVPENV VINS PAC7MX $PACTMP/WMX`
export PAC7MX
PAC7RQ=`BVPENV VINS PAC7RQ /dev/null`
export PAC7RQ
PAC7TD=`BVPENV VINS PAC7TD $PACUSERS/TD.txt`
export PAC7TD
PAC7WD=`BVPENV VINS PAC7WD $PACTMP/WWD`
export PAC7WD
SYSEXT=`BVPENV VINS SYSEXT $PACTMP/WSY`
export SYSEXT
BVPMSG 1009 "BVPVINS"
rtspac BVPVINS
RETURN=$?
case $RETURN in
0)
;;
4)
BVPMSG 1008 $PACUSERS
;;
*)
BVPMSG 1012 "BVPVINS"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

---

## GS/HE/ZS Files Initialization Utilities

### LDGS - Introduction

The LDGS procedure enables to physically create and initialize the GS indexed file.

This procedure must be executed when the GS file is not physically created (during the first installation especially) or logically disorganized.

#### Execution condition

On-line servers must be shut down.

### User input

None.

## **LDGS - Description of Steps**

### Initialization of the GS file: PTLDGS

This step initializes the GS file with a record.

Code	Physical name	Type	Label
PAC7GS	Database dir.: GS	Output	Database work file

## **LDGS - Execution Script**

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) LDGS BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      INITIALIZATION OF THE GS FILE
# *
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "LDGS"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
echo "-----"
echo ""
BVPPAUSE
# -----
. $PACDIR/config/$1/PAC7GS.ini
BVPMSG 1009 "BVPTLDGS"
rtspac BVPTLDGS
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTLDGS"
BVPERR
exit $RETURN
;;
```

```

esac
# -----
BVPMSG 1010
exit $RETURN

```

## LDHE - Introduction

The LDHE procedure enables to physically create and initialize the HE indexed file.

This procedure must be executed when the HE file is not physically created (during the first installation especially) or logically disorganized.

### Execution condition

On-line servers must be shut down.

### User input

None.

## LDHE - Description of Steps

### Initialization of the HE file: PTLDHE

This step initializes the HE file with a record.

Code	Physical name	Type	Label
PAC7HE	Base dir.: HE	Output	Backup file before display of Help documentation

## LDHE - Execution Script

```

#!/bin/sh
#@(#)VA Pac xxx xxx (R) LDHE BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "LDHE"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
echo "-----"
echo ""
BVPPAUSE
# -----
. $PACDIR/config/$1/PAC7HE.ini
BVPMSG 1009 "BVPTLDHE"

```

```

rtspac BVPTLDHE
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTLDHE"
BVPERR
exit $RETURN
;;
esac
# -----
BVPMSG 1010
exit $RETURN

```

## LDZS - Introduction

This procedure enables to physically create and initialize the ZS indexed file.

It must be executed when the ZS file is not physically created (during the first installation especially) or logically disorganized.

### Execution condition

On-line servers must be shut down.

### User input

None.

## LDZS - Description of Steps

### Initialization of the ZS file: PTLDTS

This step initializes the ZS file with a record.

Code	Physical name	Type	Label
PAC7ZS	Base dir.: ZS	Output	Database work file

## LDZS - Execution Script

```

#!/bin/sh
#@(#)VA Pac xxx xxx (R) LDZS BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "LDZS"
echo "===== "

```

```

BVPMMSG 1047 "$BVPBASE"
BVPMMSG 1005 "$PACDIR/config/$1"
echo "-----"
echo ""
BVPPAUSE
# -----
. $PACDIR/config/$1/PAC7ZS.ini
BVPMMSG 1009 "BVPTLDTS"
rtspac BVPTLDTS
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMMSG 1012 "BVPTLDTS"
BVPERR
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
exit $RETURN

```

---

## Retrieval utilities

### UTU1 UTU2 - Adaptation of 'UNS' operators

#### UTU1 - Extraction of 'P' lines with 'UNS'

##### Principle

This tool is used to extract the programs' 'P' lines which contain the 'UNS' operator and do not contain anything in the 'Level-Condition type' field. The user will have to check the output file and delete, in this file, all the lines he/she wants to keep as they are. For the other lines, the 'Level-Condition type' field will have to be forced to '99BL' before the execution of the UTU2 procedure.

##### Execution conditions

Enter the name of the result file of the lines to be checked in the startup script: NOMUT='...'

##### Result

A 'NOMUT' file which contains UNS 'P' lines to be checked.

#### UTU1 - Input / Processing / Results

No user input.

## UTU1 - Description of Steps

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

Code	Physical name	Type	Label
PAC7PC	Save dir.: PC	Input	Sequential image of the Development Database
UTUTO	User dir.: UTUTI	Output	'P' lines with 'UNS' to be updated

## UTU1 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTU1 BATCH PROCEDURE
# * -----
# *          VISUALAGE PACBASE
# *
# * -----
# *          EXTRACTION OF LINES "P"
# *          WITH OPERATOR "UNS"
# *          WITHOUT LEVEL-CONDITION TYPE
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTU1"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
UTUTO=`BVPENV PTUNS1 UTUTO `dirname $PACUSERS`/UTUTI`
export UTUTO
. $PACDIR/config/$1/PACSAVPC.ini
PAC7PC=`BVPENV PTUNS1 PAC7PC $PACSAVPC`
export PAC7PC
BVPMSG 1009 "BVPTUNS1"
rtspac BVPTUNS1
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUNS1"
BVPERR
BVPRMTMP
```



```

        exit $RETURN
    ;;
esac
# -----
BVPMSG 1010
BVPRMTMP
exit $RETURN

```

## UTU2 - Update of 'P' lines with 'UNS'

### Principle

This tool is used to update the transactions extracted with the UTU1 procedure.

For all the lines in the input file, the 'Level- Condition type' area is forced to '99BL'.

### Execution conditions

Specify the result file name of the UTU1 procedure in the startup Script:  
 NOMUT='... '

### Result

A new sequential image of the Development Database.

## UTU2 - Input / Processing / Results

No user input.

## UTU2 - Description of Steps

Extracting 'P' lines with 'UNS' operator: UTIUN2

Code	Physical name	Type	Label
PAC7PC	Save dir.: PC	Input	Sequential image of the Development Database
UTUTI	User dir.: UTUTI	Input	'P' lines with 'UNS' to be updated
PAC7CP	Save dir.: PC-new	Output	New sequential image of the Development Database

## UTU2 - Execution Script

```
#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTU2 BATCH PROCEDURE
# * -----
# *      VISUALAGE PACBASE
# *
# * -----
# *      UPDATE LINES "P"
# *      WITH OPERATOR "UNS"
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTU2"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
UTUTI=`BVPENV PTUNS2 UTUTI `dirname $PACUSERS`/UTUTI`
export UTUTI
. $PACDIR/config/$1/PACSAVPC.ini
PACSAVPC=`BVPENV PTUNS2 PAC7PC $PACSAVPC`
PAC7PC=$PACSAVPC
export PAC7PC
PAC7CP=$PACSAVPC.NEW
export PAC7CP
BVPMSG 1009 "BVPTUNS2"
rtspac BVPTUNS2
RETURN=$?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUNS2"
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMSG 1010
BVPMSG 1016 "PCBACKUP.ini"
sh $PACDIR/config/$1/PCBACKUP.ini
BVPRMTMP
exit $RETURN
```

## UTM1 UTM2 - Conversion of 'old' Meta Entities

### Principle

This procedure is a utility whose purpose is to convert the 'old' meta entities into 'formatted' meta entities.

The PC25 procedure retrieves the 2.n user entities into 3.n 'old'-type meta entities. These meta entities cannot be modified.

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

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

### Migration process

#### Step 1: Analysis of the Database relations (UTM1)

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

Any 'old'-type relation must be converted into as many relations as the number of entities they are associated with.

This first step consists in searching, in the VA Pacbase Database, the calls to 'old' relations in the description lines of the meta entities and in all the entities.

The list of calls is printed in the output file.

In this file, the '&&' characters separate the printed information from the information in transaction format.

You will have to specify the new relation code.

The old code can be kept if the relation is associated with one entity type only.

Example of transformation:

```
RELAT1 P 220 P PGMXCR CR VAO 2243 && ...
```

modified as:

```
RELATE P 220 P PGMXCR CR VAO 2243 && ...
```

The RELATE relation will then be dedicated to the 'P'-type entity (Program), whereas RELAT1 was dedicated to several entity types.

Note: Only the Relation name, located before the && characters, is to be modified. The continuation of the line must remain as is.

If the same relation code is linked to UEs, but in libraries which belong to different branches, you must rename the relation for each new library.

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

- Prior to this conversion, all the relations of the PAC7ME file are checked to ensure that they all point to only one entity type. If they do not, you are requested, via a display, to make them point to only one type and to re-run the UTM2 procedure. The procedure is then stopped.

When the ME file is consistent, the 'old'-type meta entities are converted into 'formatted' meta entities.

- The relations you have selected are modified or created.

If the relation already exists in the Database, it can be kept if it is defined in a convenient context for the calling entity, if it is 'constrained' and if the associated entity type is the same as the chosen type.

If the relation is 'old', it will be changed into 'constrained' and the chosen entity type will be associated with it.

If the existing relation is not suitable, an error message is printed and you will have to choose a new relation code.

All the relations are created in the library and the 'H' session which is the closest to their use.

- Relations calls are updated in the concerned entities (description lines of meta entities and others).
- If errors are detected (return code 8), you will have to modify the input file PAC7ME and start again the UTM2 procedure before the update. Caution, the &UTM2MV permanent file must first be deleted or renamed.

#### Update

The update is performed by the BVPACA15 program.

A backup of the Database is recommended before the update.

Only transactions where an error is detected are printed. There may be many rejects because these entities created in the 2.n version are much more controlled in the 3.n version. Manual interventions in the Database are then required.

The transactions are not journalized.

The &UTM2MV transaction file is declared as a permanent file to enable the user to view all the transactions impacted by the update.

### Step 3: Inventory of fixtures

It is recommended to re-execute the first step to make sure that the Database no longer contains calls to 'old'-type relations.

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

### Step 4: Reorganization

When the conversion is acceptable, you must reorganize the Database.

### Execution conditions

None during step 1 (UTM1).

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

### Printed output

At the end of step 1, a report is printed. It contains the list of the calls to 'old'-type relations.

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

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

### Result

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

## **UTM1 - Description of Steps**

### Analysis of the Database relations: PTUME1

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Development Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Development Database Index
PAC7ME	User dir. : UTM1ME	Output	List of relations calls in all the entities (length = 221)
PAC7AE	System - Skel. dir.: AE	Input	Error messages

Code	Physical name	Type	Label
PAC7EQ	User dir.: UTM1EQ	Report	Report

### UTM1 - Execution Script

```

#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTM1 BATCH PROCEDURE
# * -----
# *     VISUALAGE PACBASE
# *
# * -----
# *     CONVERSION OF OLD META-ENTITIES
# *     ANALYSIS OF THE DATABASE RELATIONS
# * -----
# *
# Parameter control
. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTM1"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
PAC7EQ=`BVPENV PTUME1 PAC7EQ $PACUSERS/UTM1EQ.txt`
export PAC7EQ
PAC7ME=`BVPENV PTUME1 PAC7ME `dirname $PACUSERS`/UTM1ME.txt`
export PAC7ME
BVPMSG 1009 "BVPTUME1"
rtspac BVPTUME1
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUME1"
BVPMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac

```

```
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN
```

## UTM2 - Input / Processing / Results

A '\*' line as input to the second step of the conversion (UTM2) with a user code and its password.

If the user code and password are not indicated, an error message is displayed and the procedure cannot be run.

## UTM2 - Description of Steps

Consistency check of the relations file : PTUME3

Code	Physical name	Type	Label
PAC7ME	User dir.: UTM1ME	Input	List of relation calls in all the entities

Conversion of the meta entities, the relations and their calls: PTUME2

Code	Physical name	Type	Label
PAC7AR	Admin. Database - Base dir.: AR	Input	Development Database data
PAC7AN	Admin. Database - Base dir.: AN	Input	Development Database index
PAC7ME	User dir. : UTM1ME	Input	List of relations calls in all the entities
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7MV	User dir.: UTM2MV	Output	Update transactions (length = 170)
PAC7ET	User dir.: UTM2ET	Output	Report
PAC7MB	User input	Input	user input

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	Database dir.: AR	Output	Development Database data
PAC7AN	Database dir.: AN	Output	Development Database index
PAC7AY	Database dir.: AY	Output	Development Database extension
PAC7AJ	/dev/null	Output	Development Database Journal

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGN	Admin. Database - Base dir.: AN	Input	Administration Database index
PACGGR	Admin. Database - Base dir.: AR	Input	Administration Database data
PACGGY	Admin. Database - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin. Database - Base dir.: GU	Input	Administration Database users
PAC7DC	Base dir.: DC	Input	DSMS elements file of the Development Database
PAC7ME	Tmp dir. : WME	Input	Work file
PAC7MV	User dir.: UTM2MV	Input	Update transactions
PAC7RB	Tmp dir.: WRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	Tmp dir.: WRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

### UTM2 - Execution Script

```

#!/bin/sh
#@(#)VA Pac xxx xxx (R) UTM2 BATCH PROCEDURE
# * -----
# *          VISUALAGE PACBASE
# *
# * -----
# *          CONVERSION OF OLD META-ENTITIES
# *          THE RELATIONS AND THEIR CALLS
# * -----
# *

```



```

# Parameter control

. $PACDIR/system/proc/BVPINIT.ini
echo ""
echo "-----"
BVPMSG 1004 "UTM2"
echo "          ====="
BVPMSG 1047 "$BVPBASE"
BVPMSG 1005 "$PACDIR/config/$1"
BVPMSG 1006 "$PACTMP"
BVPMSG 1073 "$PACUSERS"
BVPMSG 1007 "$PACINPUT"
echo "-----"
echo ""
BVPPAUSE
BVPMKDIR
# -----
PAC7ME=`BVPENV PTUME3 PAC7ME `dirname $PACUSERS`~/UTM1ME.txt`
export PAC7ME
BVPMSG 1009 "BVPTUME3"
rtspac BVPTUME3
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUME3"
BVPMSG 1025
BVPERR
BVPRTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
PAC7ET=`BVPENV PTUME2 PAC7ET $PACUSERS/UTM2ET.txt`
export PAC7ET
PAC7MV=`BVPENV PTUME2 PAC7MV $PACUSERS/UTM2MV.txt`
export PAC7MV
PAC7ME=`BVPENV PTUME2 PAC7ME `dirname $PACUSERS`~/UTM1ME.txt`
export PAC7ME
PAC7MB=$PACINPUT
export PAC7MB
BVPMSG 1009 "BVPTUME2"
rtspac BVPTUME2
RETURN=?
case $RETURN in
0)
;;
*)
BVPMSG 1012 "BVPTUME2"
BVPMSG 1025
BVPERR

```

```

BVPRMTMP
exit $RETURN
;;
esac
# -----
. $PACDIR/config/$1/PAC7AE.ini
. $PACDIR/config/$1/PAC7AJ.ini
. $PACDIR/config/$1/PAC7AN.ini
. $PACDIR/config/$1/PAC7AR.ini
. $PACDIR/config/$1/PAC7AY.ini
. $PACDIR/config/$1/PACGGN.ini
. $PACDIR/config/$1/PACGGR.ini
. $PACDIR/config/$1/PACGGU.ini
. $PACDIR/config/$1/PACGGY.ini
. $PACDIR/config/$1/PAC7DC.ini
. $PACDIR/config/$1/SEMLOCK.ini
PAC7IE=`BVPENV PACA15 PAC7IE /dev/null`
export PAC7IE
PAC7IF=`BVPENV PACA15 PAC7IF $PACUSERS/UTM2IFA15.txt`
export PAC7IF
PAC7ME=`BVPENV PACA15 PAC7ME /dev/null`
if [ "$PAC7ME" = "/dev/null" ]
then
    PAC7ME=$PACTMP/ME
    touch $PAC7ME
fi
export PAC7ME
PAC7MV=`BVPENV PACA15 PAC7MV $PACUSERS/UTM2MV.txt`
export PAC7MV
PAC7RB=`BVPENV PACA15 PAC7RB /dev/null`
export PAC7RB
PAC7RY=`BVPENV PACA15 PAC7RY /dev/null`
export PAC7RY
BVPMSG 1009 "BVPACA15"
rtspac BVPACA15
RETURN=$?
case $RETURN in
0)
    ;;
2)
    BVPMSG 1012 "BVPACA15"
    BVPMSG 1054
    BVPERR
    BVPRMTMP
    exit $RETURN
    ;;
4)
    BVPMSG 1012 "BVPACA15"
    BVPMSG 1055
    BVPERR
    BVPRMTMP
    exit $RETURN
    ;;
*)
    BVPMSG 1012 "BVPACA15"

```

```
BVPMMSG 1025
BVPERR
BVPRMTMP
exit $RETURN
;;
esac
# -----
BVPMMSG 1010
BVPRMTMP
exit $RETURN
```







Part Number: DELIX001372A - 9845

Printed in USA