

VisualAge Pacbase



# Installation Guide Windows 2000 or NT Server & Workstation Components

*Version 3.0*





VisualAge Pacbase



# Installation Guide Windows 2000 or NT Server & Workstation Components

*Version 3.0*

#### Note

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

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

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

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

#### Fourth Edition (August 2002)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.0

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

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

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

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

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

---

# Contents

<b>Notices</b> . . . . .	<b>v</b>	The bases.ini file . . . . .	26
<b>Trademarks</b> . . . . .	<b>vii</b>	The vaplocat.ini file . . . . .	28
<b>Chapter 1. Foreword</b> . . . . .	<b>1</b>	Uninstalling Client Components . . . . .	30
Introduction . . . . .	1	<b>Chapter 5. Tests</b> . . . . .	<b>31</b>
VisualAge Pacbase Architecture . . . . .	1	List of Utilities . . . . .	31
Contents of Supply . . . . .	2	Installation Tests . . . . .	31
Bibliography . . . . .	2	Generation-Print , TP and Batch Update Tests . . . . .	31
<b>Chapter 2. Prerequisites</b> . . . . .	<b>3</b>	Administration Database Procedures Tests . . . . .	32
Prerequisites for Server Environment . . . . .	3	Development Database Procedures Tests . . . . .	32
Hardware and Software . . . . .	3	Extraction-Utility Tests . . . . .	33
Disk Space . . . . .	3	<b>Chapter 6. Re-installation of Server</b> . . . . .	<b>35</b>
Prerequisites for Client Environment . . . . .	3	Reinstallation . . . . .	35
Hardware . . . . .	3	Uninstall the system . . . . .	35
Disk Space . . . . .	4	Data deletion . . . . .	35
Software . . . . .	4	<b>Chapter 7. Retrieval</b> . . . . .	<b>37</b>
Communication . . . . .	4	Retrieval of VisualAge Pacbase 2.0 and 2.5 . . . . .	37
<b>Chapter 3. Installation of Server Environment</b> . . . . .	<b>5</b>	Operations to be Performed . . . . .	37
Installation . . . . .	5	Retrieval of User Parameters (PE25) . . . . .	39
System Installation . . . . .	6	PE25 - Introduction . . . . .	39
Repository Installation . . . . .	6	PE25 - Input / Processing / Results . . . . .	39
Administration Database . . . . .	6	PE25 - Description of Steps . . . . .	40
Development Databases . . . . .	7	PE25 - Execution Script . . . . .	42
Create New Database . . . . .	7	Retrieval of the Development Database (PC25) . . . . .	44
Configuration parameters . . . . .	8	PC25 - Introduction . . . . .	44
Components dates list . . . . .	8	PC25 - Notes on Data Retrieval . . . . .	44
INSL - execution Script . . . . .	8	PC25 - Input / Processing / Results . . . . .	46
<b>Chapter 4. Installation/Re-installation of Client Components</b> . . . . .	<b>11</b>	PC25 - Description of Steps . . . . .	46
Things to Know Before Installing . . . . .	11	PC25 - Execution Script . . . . .	50
Fundamentals of VA Pac Client-Server Communication . . . . .	12	Generation-Print Commands Retrieval (PG20) . . . . .	52
Installation Startup . . . . .	14	PG20 - Introduction . . . . .	52
Administrator & Developer workbench . . . . .	14	PG20 - Input / Processing / Results . . . . .	53
eBusiness Tools . . . . .	17	PG20 - Description of Steps . . . . .	53
VisualAge Pacbase WorkStation . . . . .	18	PG20 - Execution Script . . . . .	57
Pacbase Web Connection . . . . .	22	Generation-Print Commands Retrieval (PG25) . . . . .	61
Middleware . . . . .	24	PG25 - Introduction . . . . .	61
Editing communication parameters . . . . .	26	PG25 - Input / Processing / Results . . . . .	61
		PG25 - Description of Steps . . . . .	62
		PG25 - Execution Script . . . . .	66
		PEI Retrieval (PP25) . . . . .	69
		PP25 - Introduction . . . . .	69

PP25 - Input / Processing / Results . . . . .	69	Server Environment Components . . . . .	91
PP25 - Description of Steps . . . . .	70	Introduction . . . . .	91
PP25 - Execution Script . . . . .	71	On-Line Documentation . . . . .	91
Retrieval of Pac/Transfer Parameters (UV25)	73	Generation Skeletons . . . . .	91
UV25 - Introduction . . . . .	73	Administration Database . . . . .	91
UV25 - Input / Processing / Results . . . . .	73	Administration Database Files . . . . .	91
UV25 - Description of Steps . . . . .	74	Administration Database Backup . . . . .	92
UV25 - Execution Script . . . . .	76	Development Database . . . . .	92
Retrieval of MB Transactions (MB25) . . . . .	78	Development Database Files . . . . .	92
MB25 - Introduction . . . . .	78	Development Database Backup Files . . . . .	93
MB25 - Description of Steps . . . . .	78	Modules - Specific Files . . . . .	93
MB25 - Execution Script . . . . .	78	Pac/Impact . . . . .	93
Retrieval of GY Transactions (GY25) . . . . .	79	DSMS . . . . .	93
GY25 - Introduction . . . . .	79	PAF . . . . .	94
GY25 - Description of Steps . . . . .	80	Complementary Libraries and Files . . . . .	94
GY25 - Execution Script . . . . .	80		
Retrieval of MB Transactions (MB30) . . . . .	81	<b>Chapter 9. Appendix . . . . .</b>	<b>95</b>
MB30 - Introduction . . . . .	81	Installation of the Administration Database	
MB30 - Description of Steps . . . . .	82	Model . . . . .	95
MB30 - Execution Script . . . . .	82	VINS - Introduction . . . . .	95
Retrieval of GY Transactions (GY30) . . . . .	83	VINS - Input / Processing / Results . . . . .	95
GY30 - Introduction . . . . .	83	VINS - Description of Steps . . . . .	96
GY30 - Description of Steps . . . . .	84	VINS - Execution Script . . . . .	97
GY30 - Execution Script . . . . .	84	Installation of the Development Database	
Retrieval of journal file PJ (PJ25) . . . . .	85	Model . . . . .	99
PJ25 - Introduction . . . . .	85	VINS - Introduction . . . . .	99
PJ25 - Description of steps . . . . .	86	VINS - Input / Processing / Results . . . . .	100
PJ25 - Execution JCL . . . . .	86	VINS - Description of Steps . . . . .	101
Procedures - Summary Table of Changes . . . . .	87	VINS - Execution Script . . . . .	102
<b>Chapter 8. Components . . . . .</b>	<b>91</b>		

---

## Notices

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

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

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

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





---

## Trademarks

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

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

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

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

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



---

# Chapter 1. Foreword

---

## Introduction

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

- Server environment,
- Client environment,
- Communication.

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

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

---

## VisualAge Pacbase Architecture

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

VisualAge Pacbase consists of:

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

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

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

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

### THE SERVER ENVIRONMENT

It consists of the following components:

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

- one or more development Database(s).

## THE CLIENT ENVIRONMENT

It consists of the following components:

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

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

---

## **Contents of Supply**

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

- Installation Guide,
- CD-Rom or cartridge, depending on the environment, to install VA Pac servers,
- Workstation Components CD-Rom,
- VA Pac documentation CD-Rom.

---

## **Bibliography**

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

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

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

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

---

## Chapter 2. Prerequisites

---

### Prerequisites for Server Environment

#### Hardware and Software

from the following URL:

- Architecture : a Windows/NT or

Windows 2000 server and Windows workstations.

- Processor: workstation with Windows NT server (4.0 minimum) or Windows 2000.
- Memory: 96 Mb. Additional memory may be needed depending on the number of servers installed on the same machine.
- Software:
  - Microsoft Windows Script, version 5.1 or higher. You can download it from the following URL:  
[www.microsoft.com/msdownload/vbscript/scripting.asp](http://www.microsoft.com/msdownload/vbscript/scripting.asp)
- Micro Focus Application Server.
- Installation support : CD-ROM drive.

#### Disk Space

The disk space taken by the files depends on the size of the applications managed by the system.

The following chart approximately shows (in million bytes) the disk space required for the installation of the servers:

Disk space Installation	Total
Total for installation	95
Total for System	59
Total for installation test user files	35
Total for installation test Administrator files	260 kb

---

### Prerequisites for Client Environment

#### Hardware

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

- Processor: Intel Pentium III 450 Mhz minimum or compatible processor.
- Monitor: graphic monitor (800x600) VGA or higher resolution (XGA or SVGA).
- CD-Rom drive.
- Card: adapted to the site network.
- Memory (RAM): 128 Mb (256 Mb advised).
- Software: Microsoft Windows Script Host (version 5.1 and onwards).

## **Disk Space**

Required disk space:

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

## **Software**

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

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

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

---

## **Communication**

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

---

## Chapter 3. Installation of Server Environment

---

### Installation

VA Pac is supplied on a CD-Rom, which contains the following elements:

- a short documentation,
- Data.Cab (any Server file in a compressed format),
- setup.exe (installation executable program),
- setup.ini (installation parameters),
- VisualAge Pacbase 3.0.Msi (installation table for Windows Installer).

The workstation on which the product is installed must have an access to the CD-Rom driver.

When you install the product, you must specify some parameters: the 'database\_name' and 'port\_number' associated with each database, the volume units (C:\ by default), as well as the files directory (\Program Files\IBM\VisualAge Pacbase by default).

To perform the installation, you must be connected with NT Administrator's authorizations.

By default, the Server is installed on

C:\Program Files\IBM\VisualAge Pacbase\

However you will be able to modify the unit and directory name during the installation process.

The installation includes three successive steps:

1. installation of the System, which creates the \SYS\ directory,
2. installation of the administration database, which creates the \CONFIG\ADMIN\ and \DATA\ADMIN\ subdirectories.
3. installation of a development database, which creates the \CONFIG\[database\_name]\ and \DATA\[database\_name]\ subdirectories.

Each step can be executed independently or not, but always in this order.

All the databases are created with the installation test data and configuration parameters.

---

## System Installation

The execution of setup.exe opens a graphical interface which guides you through the installation process.

After a Welcome screen, it displays some workstation characteristics (which can be modified): of the workstation (owner's Name and Organization).

- Owner's Name and Organization,
- Location for the installation (C:\Program Files\IBM\VisualAge Pacbase\ by default)

The installation copies:

- the translation file, the labels file produced by the procedures in case of errors or for information,
- the skeletons in the various languages of the product,
- the programs
- the procedures
- the database creation utility.

---

## Repository Installation

### Administration Database

This step creates the Administrator Database which monitors the user Databases and adds batch procedures specific to the Administration environment.

Only one Administration Database can be installed. Its name is ADMIN.

The following elements must be specified:

- the VA Pac access key (70 char.)
- the Database port number (between 6000 and 9000)
- the Database language code (English by default)

You can modify the default directories (C:\Program Files\IBM\VisualAge Pacbase\).

The following procedures are executed: INAE, INGU , REST , VINS.

**Note::** You can install the Administration Database just after the system installation, or later on by clicking the [CREATE NEW DATABASE] icon.



## Development Databases

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

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

### NOTE

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

The installation of a new database creates the files used by the Developer to create and maintain entities in the Dictionary.

More than one development Database can be installed. Each has its own environment.

You must specify the following elements:

- the Database name (8 char.)
- the Database code (4 char.), as it is declared in the Administration Database.
- the Database port number (between 6000 and 9000)  
Warning: it must be different for each database.
- the Database port number (between 6000 and 9000, unique for each Database).
- the Database language code (English by default)
- the generation Language code (English by default)

You can modify the proposed directories (by default: C:\Program Files\IBM\VisualAge Pacbase\).

The following procedures are executed: REST, VINS.

## Create New Database

You can use the [CREATE NEW DATABASE] icon to install the Administration Database or a Development Database at any time after the system installation.

The installation process requires a user code with a level 4 authorization. In the test Administration Database, the 'ADMIN' user, with a level 4 authorization, is already declared.

## Configuration parameters

For each database, a Configuration directory is created with

- 1- BvpServer.ini, containing the parameters of the conversational server,
- 2- Fhredir.cfg , containing the "Rollback" parameters.

---

## Components dates list

### INSL procedure

This procedure is used to edit the following lists:

- the list of batch, on-line 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.

## INSL - execution Script

```
REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      - LIST OF INSTALLED PROGRAMS AND FILES -
REM *
REM * -----
REM *
REM *
REM * INPUT:
REM * - ONE LINE : BVP PREFIX OF THE PRODUCT
REM *   (COL.3 LENGTH 3 )
REM *
REM * -----
REM *

Dim fso, dir, subdir, file, PathFile
Dim ListExclude
Dim regEx, retVal

Set WshShell = Wscript.CreateObject("Wscript.Shell")

' Set Path of programs :
' -----
Reg_PROD = "HKLM\SOFTWARE\IBM\BVP VisualAge Pacbase\"
Rep_INST = WshShell.RegRead (Reg_PROD)

Rep_PGM = Rep_INST & "Server\Sys\pgm"
Rep_CFG = Rep_INST & "Server\Config"

' Set List of cobol programs :
' -----
ListCobol = Rep_CFG & "\" & "ListCob.txt"
Set fso = WScript.CreateObject("Scripting.FileSystemObject")
```

```

Set dir = fso.GetFolder(Rep_PGM)
Set StreamLst = fso.CreateTextFile(ListCobol, true, false)
Set regEx = New RegExp

ListExclude = " BVPSEVER.EXE BVPDIAL.EXE BVPWSH.OCX "
ListExclude = " BVPLOCK.DLL BVPPAUSE.DLL      " & ListExclude
ListExclude = " BVPSPLIT.EXE BVPTRANS.EXE    " & ListExclude
ListExclude = " CODEPAGECONV.DLL VAPUTIL.DLL " & ListExclude
ListExclude = " BVP.ICO BVPSEVER.ICO         " & ListExclude

For Each file In dir.Files
    regEx.Pattern = file.Name
    regEx.IgnoreCase = True
    retVal = regEx.Test(ListExclude)
    'Cobol program
    If retVal = False Then
        LineLst = Rep_PGM & "\" & file.Name & vbCRLF
        StreamLst.Write(LineLst)
    End If
Next
Set StreamLst = Nothing

' Launch INSL :
' -----
' Debug Mode
' CREATE_LOG = True
CREATE_LOG = False

Set WshEnv = WshShell.Environment("PROCESS")
WshEnv("LIST") = ListCobol

INSL_LOG=chr(34) & Rep_CFG & "\BVPINSL.log" & chr(34)
INSL_OUT=chr(34) & Rep_CFG & "\BVPINSL.txt" & chr(34)
sCmd = Rep_PGM & "\Bvupdate.exe " & INSL_OUT
If ( CREATE_LOG = True ) Then
    sCmd = "cmd.exe /c " & sCmd & " 1>" & INSL_LOG & " 2>&1"
    Return = WshShell.Run( sCmd, 7 , TRUE)
Else
    Return = WshShell.Run( sCmd, 10 , TRUE)
End If

' End of job :
' -----
If ( Return <> 0 ) Then
    Display = "Abnormal end of Job."
else
    ' delete list of cobol programs
    Set FileD = fso.GetFile(ListCobol)
    FileD.Delete
    Display = "End of Job." & vbCRLF & vbCRLF & _
    "File " & INSL_OUT & " has been created."
End If

If ( CREATE_LOG = True ) Then
    Display = Display & vbCRLF & vbCRLF & _

```

```
        "File " & INSL_LOG & " has been created."  
    End If  
  
    WScript.Echo(Display)  
    Wscript.Quit (0)
```

---

## Chapter 4. Installation/Re-installation of Client Components

---

### Things to Know Before Installing

- To install the VisualAge Pacbase Client components on a Windows workstation, you must have an Administrator profile on that Workstation.
- VA Pac Client components are installed via InstallShield for Windows Installer (ISWi).  
If Windows Installer is not installed on the workstation, it will be installed automatically.
- You also need Microsoft Windows Script, version 5.1 or higher. You can download it from the following URL:  
[www.microsoft.com/msdownload/vbscript/scripting.asp](http://www.microsoft.com/msdownload/vbscript/scripting.asp)
- Both Administrator & Developer workbench and eBusiness Tools components require, for their online help, that Netscape or Internet Explorer 5.5 be installed.
- The installation of a Client component does not require the prior installation on the server of the VA Pac Database(s) to which it will connect. However, the code of each VA Pac Database you indicate when you install a Client Component will have to be strictly reused when these Databases are installed at the server level.
- By default, the root directory of all the VisualAge Pacbase Client components is:

C:\Program Files\IBM\VisualAge Pacbase\

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

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

However, for the installation of a later version of a component the installer will propose:

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

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

C:\Program Files\IBM\VisualAge Pacbase\_1\

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

C:\Program Files\IBM\VisualAge Pacbase\_2\

## Fundamentals of VA Pac Client-Server Communication

**FOREWORD:** The following lines present the principles of communication between the Client components and the VisualAge Pacbase server. These lines contain important information essential to the choice of communication protocol and the parameterization of the associated middleware. This information will also be useful for future installations (other Client components or new versions of already installed components).

### AVAILABLE COMMUNICATION TYPES

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

**NOTE ::**

### THE MIDDLEWARE

The middleware must always be installed on each Developer workstation. This installation starts automatically during the first installation under a given root of one of the following Client components: Administrator and Developer workbench, VisualAge Pacbase Workstation, eBusiness Tools and Pacbase Web Connection. For the latter, this installation is automatically launched if the Context Server option is selected.

The middleware installed on each Developer workstation then ensures direct communication between the Client component(s) and the Server.

However you can also choose a communication via a gateway.

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

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

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

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

### COMMUNICATION FILES

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

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

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

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

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

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

### THE 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.

### 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.

## Installation Startup

Insert the installation CD.

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

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

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

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

Choose the Client component you want to install.

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

---

## Administrator & Developer workbench

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

In the next panel, select one or more elements to install:

- Administrator workbench

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

- Developer workbench

Developer workbench includes the following modules:



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

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

- Rational Rose Bridge.

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

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

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

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

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

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

To do this, a window enables you to enter:

1. The name of a Database, already installed at the server level or not. The names entered here will be displayed in the connection smartguide, thus showing users which Databases they can connect to. The name given to each Database should therefore be clear enough to be easily identified in the list of Databases proposed by the connection smartguide.
2. The Database logical code.  
Maximum length: 4 characters.

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

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

3. The location name

Maximum length: 20 characters.

Default: Location-1

More than one Database may be associated with one location.

4. Finally, specify the user authentication mode at connection. Refer to the Database Administrator to ensure authentication measures at the server level are imposed at the workbench level. You select the mode via two check boxes.

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

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

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

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

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

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

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

### START-UP FILES

The start-up files are :

admin.bat

batch.bat

eBusiness.bat

runcfm.bat

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

## START MENU / PROGRAMS CHOICE

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

Administrator-Developer workbench  
Administration  
Batch  
eBusiness

---

### **eBusiness Tools**

The eBusiness Tools are:

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

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

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

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

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

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

The eBusiness Tools component can run via a gateway. In this case you will have to install the Middleware component and configure the communication at the level of the intermediate server which hosts the VisualAge Pacabase gateway.

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

The middleware installed in this context allows communication between the server and the generated proxies.

Communication parameters will have to be set by the developer with the Location Editor tool included in this installation.

### START-UP FILES

The start-up files are :

- For the Proxy Generator:  
vapgen.exe
- For the Location Editor:  
vapLocationEditor.exe
- For the Services Test Facility:  
vapServicesTestFacility.exe

These files are to be found in the eBusiness Tools root directory (EBusinessTools).

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

### START MENU / PROGRAMS CHOICE

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

```
eBusinessTools
    Location Editor
    Proxy Generator
    Services test Facility
```

---

## **VisualAge Pacbase WorkStation**

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

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

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

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

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

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

In the next panel, select the elements to install:

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

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

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

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

- If you choose the direct communication option, the middleware installation script will automatically start once the installation of the workstation is finished. It will require the configuration of communication parameters. For information on this part the installation, see the subchapter Middleware.
- If you choose the gateway option, enter the IP address of the gateway here. The installation of the middleware on the Developer workstation - also necessary in this context - will then start automatically after the installation of the WorkStation.

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

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

To do this, a panel enables you to enter:

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

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

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

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

2. The Database logical code.

Maximum length: 4 characters.

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

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

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

3. The location name

Maximum length: 20 characters.

Default: Location-1

More than one Database may be associated with one location.

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

You select the mode via two check boxes.

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

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

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

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

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

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

### START-UP FILE

The start-up file is :

pexec.exe

This file is to be found in the VisualAge Pacbase WorkStation root directory (SPAC).

### START MENU / PROGRAMS CHOICE

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

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

### INSTALLATION PARAMETERS FILE

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

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

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

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

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

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

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

## DESCRIPTION OF THE PARAMETERS FILE

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

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

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

```
001 Station Version          [300F]
002 Server                   [PACBASE]
003 Communication Manager    [MWC0M]
004 Communication Parameters [MWC0M]
005 System                   [WIND0WS]
006 Method                   [MER]
007 EXE disk                 [C]
008 EXE disk(default)       [C]
009 System Data Disk        [C]
010 User Data Disk          [C]
011 Connection execution mode [E]
```

The Pacbase.dat file should not be destroyed.

The possible values for the Methodology parameter are:

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

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

---

## **Pacbase Web Connection**

### INSTALLATION PREREQUISITE

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

<http://www.perl.com>



To install the interpreter, follow the indications given below.

## INSTALLATION

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

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

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

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

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

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

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

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

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

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

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

## EXECUTION FILE

The Pacbase Web Connection execution file is:

Pacweb.exe

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

### INSTALLATION OF THE CONTEXT SERVER AS AN NT SERVICE

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

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

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

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

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

When the computer boots up, it is automatically executed.

To uninstall the service, enter the following command:

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

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

---

## **Middleware**

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

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

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

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

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

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

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

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

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

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

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

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

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

#### LIST OF PARAMETERS

- VAP SOCKET  
IP address: IP address and port used by the VA Pac server
- MVS CICS SOCKET  
IP address: IP address and port used by the VA Pac server  
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/IMS CPI-C  
Destination-id entry: BVPSCPI (default value). If you modify this value, it must be the same as the value entered in the Symbolic destination name, a parameter included in the configuration of this communication protocol.  
Protocol code page: Value identifying the coding of characters used by the VA Pac server. 1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MVS IMS Connect

IP address: IP address and port used by the VA Pac server.

Transaction code : IMS transaction code of the VA Pac Communications Monitor.

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

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

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

---

## 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\Common\

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

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

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

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

**NOTE:** With the VA Pac WorkStation, you may use a personalized parameters file. To do so, enter -- after the VA Pac Database name -- this file's name framed by the "<" and ">" signs.

Complete details on these parameters are given above, at the end of the 'VisualAge Pacbase WorkStation' subchapter.

## DESCRIPTION OF A SECTION'S CONTENTS

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

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

Maximum length: 4 characters

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

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

The following parameters vary according to the chosen option.

### PARAMETERS FOR DIRECT ADAPTER (LOCAL MIDDLEWARE)

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

Default: ..\common\vaplocat.ini

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

- location = location name for the Database

Maximum length: 20 characters.

Default: Location-1

More than one Database can point to the same location.

### REMINDER

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

- traceFile = path and name of the file which will receive the trace of the middleware execution.

By default this file is automatically created (with timestamp) in the VapTrace sub-directory.

- traceLevel = trace level of the middleware execution. Its possible values are:
  - 0 : no trace
  - 1 : trace with errors (default)
  - 2 : standard trace, not detailed
  - 3 : trace for information
  - 4 et + : trace for debug
- codePageFile = path and name of the file which contains the conversion table of the code pages.  
Default: ..\middleware\CharConv.txt

#### PARAMETERS FOR GATEWAY ADAPTER (REMOTE MIDDLEWARE)

- host = name or IP address of the host where the VisualAge Pacbase gateway is installed.  
Default: 127.0.0.1 for a local host
- port = value of the IP port where the gateway receives the client requests.  
Default: 5647
- location = location name for the Database  
Maximum length: 20 characters.  
Default : Location-1  
More than one Database can point to the same location.

#### REMINDER

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

### **The vaplocat.ini file**

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

By default, the path to this file is:

C:\Program Files\IBM\VisualAge Pacbase\Common\

**NOTE:** All parameters in the vaplocat.ini file are not explained here. In fact, certain parameters allow finer middleware settings, particularly used by proxies (generated by eBusiness Tools). In this context, these parameters are edited with the Location Editor tool and are therefore documented in its online help.

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

The maximum length of this name is 20 characters.

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

- VAP SOCKET

```
<LocationName>
COMM_TYPE=SOCKET
MONITOR=BVPSCPI
MESSAGE_LENGTH=31744
IXO_TIMEOUT=30
IXO_ADDRESS=127.0.0.1 3000
```
- 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 3000
IXO_TRANSID=V303
```
- MVS CICS/IMS CPI-C

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

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

#### DETAILS ON THE PARAMETERS

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

- COMM\_TYPE:  
This parameter identifies the communication protocol in use.

The possible values are:

SOCKET: VA Pac Server under Windows or UNIX, with the use of TCP/IP.  
TCPMVS: VA Pac Server under MVS/CICS or MVS/IMS, with the use of a TCP/IP listener.

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

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

- **IXO\_ADDRESS:**

IP address and port used by the VA Pac Server. The port number must correspond to the one indicated at the server configuration.

- **IXO\_DATASTORE:**

Name of link to IMS defined in IMS Connect (IMS DataStore ID)

- **IXO\_RACFGROUP:**

Name of RACF group for IMS Connect.

- **IXO\_TIMEOUT:**

Maximum time required for a workstation to receive an answer from the server before indicating any communication error.

This parameter is indicated in seconds. Its default value is 30.

- **IXO\_TRANSID:**

CICS transaction code

- **HOST\_ENCODING:**

Identifies the encoding of the characters used by the VisualAge Pacbase server.

1140 (US EBCDIC) or 1146 (UK EBCDIC)

- **MESSAGE\_LENGTH:**

This parameter's value must be 31744.

- **MONITOR:**

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

**NOTE:**

---

## Uninstalling Client Components

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

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



---

## Chapter 5. Tests

---

### List of Utilities

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

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
ARCH	Saving of the Development Database journal
PACS	Saving of the Development Database (SAVE)
PACS	Library management of the Development Database (MLIB)
REOR	Reorganisation 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 on-line and batch update tests,
- Administration procedures tests,
- Extraction utility tests.

#### Generation-Print , TP and Batch Update Tests

These tests consist of the following steps:

- On-line use tests:
  - 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**

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.
- Saving of the Administration Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file.
- 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 Database backup file: execute the REST procedure.

The Administration Database server must be shut down while these tests are being performed.

### **Development Database Procedures Tests**

The tests on the procedures of the Development Database 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.
- Direct backup of the Development Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file.
- Library management: add/delete a Library in the Development Database: execute the PACS procedure (MLIB option), 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 PACX procedure (EXPJ option).

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

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



---

## Chapter 6. Re-installation of Server

The system environment of the VisualAge Pacbase server must be re-installed whenever enhancements to the currently installed version are being delivered.

The Windows NT service -Add/remove programs- enables you to do it with the 'Add/remove' options.

---

### Reinstallation

- The updating of the system is executed automatically if you have an older version already installed.

---

### Uninstall the system

You uninstall the system via the 'Add/remove programs' service of the control panel ('VisualAge Pacbase Server' application).

You are advised to remove all data accesses from the Register database at the same time but to save the data in another directory \DATA.

---

### Data deletion

Deleting a Database is an operation that cannot be undone. It is then advised to save the Database first.

The deletion is performed via manual operations:

- Deletion of the directory which contains the Database data, and all its dependent sub-directories,
- Deletion of the directory:

Winnt\Profiles\AllUsers\Start Menu\Programs\VisualAge  
Pacbase Server\[database\_name]

- Deletion from the HK\_Local\_Machine register of the key:  
Software\IBM\BVP VisualAge Pacbase\Server\BVP\_DATA\[database\_name].



---

## Chapter 7. Retrieval

---

### Retrieval of VisualAge Pacbase 2.0 and 2.5

#### Operations to be Performed

The installation of the 3.0 release requires, in the one hand, the retrieval of the AG (generation-print commands file), AE AP (user parameters files) and AB AC (PEI files) files in the new Administration Database, and on the other hand, the retrieval of the old Development Database.

#### SCREEN BRANCHING:

It consists of six steps:

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

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

2) Installation of the 3.0 Administration Database

To install the Administration Database, execute the installation process.

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 (enter the access key),
  - VINS: installation of the Administration Model (see the Appendix, at the end of the manual),
- re-organization of the Administration Database if an end-of-installation message requests it:
  - PACS (SAVE option): 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,
  - 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 new Development Database,
- REST: restoration of the new Development Database from the backup obtained previously,
- VINS: installation of the new Database development Model (see the description at the end of the manual).

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

- 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 and 6 are optional.

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

It consists in executing the following procedures:

- PG20: if retrieval of 2.0 AG file,
- PG25: if retrieval of 2.5 AG file.

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

It consists in executing the following procedure:

- UV25: retrieval of the UV file data.

### 6) Retrieval of PEI files.

It consists in executing the following procedure:



- PP25: retrieval of the PP file data.

---

## Retrieval of User Parameters (PE25)

### PE25 - Introduction

#### PRINCIPLE

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

#### EXECUTION CONDITIONS

The administration database files must be closed to on-line use.

#### PRINTED OUTPUT

This procedure prints a report which indicates the errors encountered.

#### RESULT

This procedure integrates the 2.0 or 2.5 user parameters in the administration database.

### PE25 - Input / Processing / Results

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

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

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

The 'A' line has the following structure:

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

A 'B' line by database in which you indicate the characteristics of the development Databases which are to be managed in the new Administration Database. You must specify:

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

- the Database name
- 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

## PE25 - Description of Steps

### PROCESSING OF USER PARAMETERS (PE): PTU920

Code	Physical name	Type	Label
PAC7EN	Save dir.: PE25	Input	User parameters, old version
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7MB	User input	Input	User input
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Base - 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. Base - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Base - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database data

Code	Physical name	Type	Label
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGY	Input	Update transactions
PAC7MV	Tmp dir.: WMV	Output	Transactions formatting (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	Working file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Working file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Working file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Working file (length=743)

#### UPDATE OF THE ADMINISTRATION DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	Admin. Base - Base dir.: AR	Output	Administration Database Data file
PAC7AN	Admin. Base - Base dir.: AN	Output	Administration Database Index file
PAC7AY	Admin. Base - Base dir.: AY	Output	Administration Database extension
PAC7AJ	Admin. Base - Base dir.: AJ	Output	Administration Database journal
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database Index file
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data file
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7DC	Base dir.: DC	Input	DSMS file of Development Database elements

Code	Physical name	Type	Label
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	nul	Output	UPDT erroneous transactions (length=80)
PAC7RY	nul	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

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

Return codes:

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

## PE25 - Execution Script

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PE FILE
REM *
REM * -----
REM *
<job id=PE25>

<script language="VBScript">
Dim MyProc
MyProc = "PE25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

```

```

Call Msg_Log (Array("1022" , "PTU920"))
'-----
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU920","PAC7EN",Rep_SAVE & "\PE.old")
Call BvpEnv("PTU920","PAC7ET",Rep_USR & "\PE25ET920.txt")
Call BvpEnv("PTU920","PAC7GY",Rep_TMP & "\WGY.tmp")
Return = WshShell.Run("BVPTU920.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU920")

```

```

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

```

```

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PE25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PE25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

```

```

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP )

```

```
Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>
```

---

## Retrieval of the Development Database (PC25)

### PC25 - Introduction

#### PRINCIPLE

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

#### EXECUTION CONDITIONS

None

#### PRINTED OUTPUT

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

#### RESULT

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

### PC25 - Notes on Data Retrieval

#### SPLITTING UP OF THE COMMENT DESCRIPTION (-G)

The comment description is split up into several descriptions.

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

#### CAUTION

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

- Generation lines

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

- Generation parameters

They include the O line type (-GO).

- Error messages management

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

- Call of entities via Relations

They include the R line type (-CR).

- Specificity of the Input Aid entity

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

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

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

#### DATA STRUCTURES TABLE TYPE

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

#### TRANSFORMATION OF U TYPE MANUALS

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

#### LONG DATA: USER ENTITIES, INPUT AIDS, REPORT LAYOUTS

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

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

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

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

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

### PC25 - Input / Processing / Results

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

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

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

This line must be structured in this way:

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

### PC25 - Description of Steps

GENERAL PROCESSES: PTU911

Code	Physical name	Type	Label
PAC7MC	Save dir. : PC25	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	Firtst data record (length=153)
PAC7PE	Tmp dir. : WPE	Output	User Entity Occurence definition (2.5 release), Report layouts, and Comments (except the calls of Input Aids) (length=193)



Code	Physical name	Type	Label
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 Occurences (2.5 release)
PAC7PF	Tmp dir. : WPF	Output	Other records (length=153)
PAC7PM	Tmp dir. : WPM	Output	Report file (length=48)
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
PAC7AE	System - Skel dir. : AE	Input	Error messages
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)

## META-ENTITIES PROCESSING: PTU912

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

## REPORT LAYOUT PROCESSING: PTU919

<b>Code</b>	<b>Physical name</b>	<b>Type</b>	<b>Label</b>
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7PE	Tmp dir. : WPE	Input	User Entity Occurences Definition (2.5 release), Report layouts and Comments (except calls of input aids)
PAC7PB	Tmp dir. : WPC	Input	First data record
PAC7PH	Tmp dir. : WPH	Input	Description of Input Aids and of their calls in the Comments
PAC7PM	Tmp dir. : WPM	Input/Output	Report file
PAC7ZP	Tmp dir. : WEP	Output	User entity Occurences Definition (2.5 release), Report layouts, and Comments (call of Input Aids included) (length=193)
PAC7PO	Tmp dir. : WPO	Output	Comments (except the call of Input Aids) (length=153)
PAC7PD	Tmp dir. : WPD	Output	First data record (length=153)

## USER ENTITIES PROCESSING: PTU913

<b>Code</b>	<b>Physical name</b>	<b>Type</b>	<b>Label</b>
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7PX	Tmp dir. : WEP	Input	Definition of User Entity Occurences (2.5 release), Report layouts, and Comments (including the call of Input Aids)

Code	Physical name	Type	Label
PAC7PZ	Tmp dir. : WZP	Input	Definition and Description of the Development Model and Description of User Entity Occurrences (2.5 release)
PAC7PB	Tmp dir. : WPD	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. : WPB	Output	First data record (length=153)

#### Sort of long data: PTU91A

Code	Physical name	Type	Label
PAC7PZ	Tmp dir. : WZX	Input	Intermediate long 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
PAC7PO	Tmp dir. : WPO	Input	Comments (no call of Input Aids)
PAC7PD	Tmp dir. : WPD	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 (3.0 release)
PAC7ET	User dir. : PC25ET914	Report	Retrieval report

## PC25 - Execution Script

```
REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PC FILE
REM *
REM * -----
REM *
<job id=PC25>

<script language="VBScript">
MyProc = "PC25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

If base = "ADMIN" Then
Call Msg_Log (Array("1028",base))
Wscript.Quit (0)
End If

Call Msg_Log (Array("1022" , "PTU911"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU911", "PAC7MC", Rep_SAVE & "\PC25")
Call BvpEnv("PTU911", "PAC7PB", Rep_TMP & "\WPC.tmp")
Call BvpEnv("PTU911", "PAC7PE", Rep_TMP & "\WPE.tmp")
Call BvpEnv("PTU911", "PAC7PF", Rep_TMP & "\WPF.tmp")
Call BvpEnv("PTU911", "PAC7PG", Rep_TMP & "\WPG.tmp")
Call BvpEnv("PTU911", "PAC7PL", Rep_TMP & "\WPL.tmp")
Call BvpEnv("PTU911", "PAC7PZ", Rep_TMP & "\WPZ.tmp")
Call BvpEnv("PTU911", "PAC7PM", Rep_TMP & "\WPM.tmp")
Call BvpEnv("PTU911", "PAC7ET", Rep_USR & "\PC25ET911.txt")
Return = WshShell.Run("BVPTU911.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU911")

Call Msg_Log (Array("1022" , "PTU909"))
'-----
Call BvpEnv("PTU909", "PAC7PI", Rep_TMP & "\WPI.tmp")
Call BvpEnv("PTU909", "PAC7PM", Rep_TMP & "\WPM.tmp")
Return = WshShell.Run("BVPTU909.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU909")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WPL.tmp")
```

```

Call Msg_Log (Array("1022" , "PTU92A"))
'-----
Call BvpEnv("PTU92A","PAC7PE",Rep_TMP & "\\WPH.tmp")
Call BvpEnv("PTU92A","PAC7PM",Rep_TMP & "\\WPM.tmp")
Return = WshShell.Run("BVPTU92A.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU92A")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPG.tmp")

Call Msg_Log (Array("1022" , "PTU912"))
'-----
Call BvpEnv("PTU912","PAC7ZP",Rep_TMP & "\\WZP.tmp")
Return = WshShell.Run("BVPTU912.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU912")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPZ.tmp")

Call Msg_Log (Array("1022" , "PTU919"))
'-----
Call BvpEnv("PTU919","PAC7PD",Rep_TMP & "\\WPD.tmp")
Call BvpEnv("PTU919","PAC7PE",Rep_TMP & "\\WPE.tmp")
Call BvpEnv("PTU919","PAC7PH",Rep_TMP & "\\WPH.tmp")
Call BvpEnv("PTU919","PAC7PO",Rep_TMP & "\\WPO.tmp")
Call BvpEnv("PTU919","PAC7ZP",Rep_TMP & "\\WEP.tmp")
Return = WshShell.Run("BVPTU919.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU919")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPC.tmp")
Call DelFile (Rep_TMP & "\\WPE.tmp")
Call DelFile (Rep_TMP & "\\WPH.tmp")

Call Msg_Log (Array("1022" , "PTU913"))
'-----
Call BvpEnv("PTU913","PAC7PB",Rep_TMP & "\\WPD.tmp")
Call BvpEnv("PTU913","PAC7PD",Rep_TMP & "\\WPB.tmp")
Call BvpEnv("PTU913","PAC7PX",Rep_TMP & "\\WEP.tmp")
Call BvpEnv("PTU913","PAC7PZ",Rep_TMP & "\\WZP.tmp")
Call BvpEnv("PTU913","PAC7ZP",Rep_TMP & "\\WZX.tmp")
Return = WshShell.Run("BVPTU913.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU913")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\\WPD.tmp")
Call DelFile (Rep_TMP & "\\WEP.tmp")
Call DelFile (Rep_TMP & "\\WZP.tmp")

Call Msg_Log (Array("1022" , "PTU91A"))
'-----

```

```

Call BvpEnv("PTU91A", "PAC7PZ", Rep_TMP & "\WZX.tmp")
Call BvpEnv("PTU91A", "PAC7ZP", Rep_TMP & "\WXZ.tmp")
Return = WshShell.Run("BVPTU91A.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU91A")

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WZX.tmp")

Call Msg_Log (Array("1022" , "PTU914"))
'-----
WshEnv("PAC7PC") = Rep_SAVE & "\PC.new"
Call BvpEnv("PTU914", "PAC7ET", Rep_USR & "\PC25ET914.txt")
Return = WshShell.Run("BVPTU914.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU914")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1022" , "BACKUP"))
'-----
Call Turnover(Rep_SAVE & "\PC")

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

---

## Generation-Print Commands Retrieval (PG20)

### PG20 - Introduction

#### PRINCIPLE

The PG20 procedure retrieves the 2.0 release PG file, sequential image of the generation-print commands, in the 3.0 release new format.

It updates the development Database with the generation-print commands and the Administration Database with the 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 in the on-line mode.

#### PRINTED OUTPUT

This procedure outputs a report which contains the errors encountered.

#### NOTE

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

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

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

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

User codes present in the PG file and not present in the administration database are automatically created for users who have 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
22	4	cccc	Database code

### **PG20 - Description of Steps**

#### GENERATION-PRINT COMMANDS FORMATTING: PTU908

Code	Physical name	Type	Label
PAC7IN	Save dir. : PG20	Input	Generation-print commands, old release
PAC7OU	Save dir. : OLDGP	Output	Re-formatted generation-print commands (length=150)

## GENERATION-PRINT COMMANDS PROCESSING: PTU921

Code	Physical name	Type	Label
PAC7PG	Save dir. : OLDPG	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. : PGnnET921	Report	Error report ( nn= 20 ou 25 , version number)

## TRANSACTIONS FORMATTING: PAF900

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Input	Development Database data
PAC7AN	Base dir.: AN	Input	Development Database index
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp. dir.: WGZ	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	Working 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 DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database Data file
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. Base - Base dir.: AN	Input	Administration Database Index file
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data file
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7DC	nul	Input	DSMS file of Development Database Elements
PAC7ME	Tmp dir.: WME	Input	Working file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	nul	Output	UPDT erroneous transactions (length=80)
PAC7RY	nul	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

TRANSACTION FORMATTING: PAF900

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

UPDATE OF THE ADMINISTRATION DATABASE: PACA15

<b>Code</b>	<b>Physical name</b>	<b>Type</b>	<b>Label</b>
PAC7AR	Admin. Base - Base dir.: AR	Output	Administration Database Data file
PAC7AN	Admin. Base - Base dir.: AN	Output	Administration Database Index file
PAC7AY	Admin. Base - Base dir.: AY	Output	Administration Database extension
PAC7AJ	Admin. Base - Base dir.: AJ	Output	Administration Database journal
PAC7AE	System - Skel. dir.: AE	Input	Error messages

Code	Physical name	Type	Label
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database Index file
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data file
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7DC	Base dir.: DC	Input	DSMS file of Development Database elements
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	nul	Output	UPDT erroneous transactions (length=80)
PAC7RY	nul	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

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

Return codes:

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

## PG20 - Execution Script

```

REM * -----
REM *     VISUALAGE PACBASE
REM *
REM * -----
REM *           RETRIEVAL OF PG FILE SINCE 2.0
REM *
REM * -----
REM *
<job id=PG20>

<script language="VBScript">
Dim MyProc

```

```

MyProc = "PG20"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

'-PG- Retrieval for the Aministration database and
'the Development database
'THUS the same steps are executed is this procedure :
' = PTUBAS + ... PACA15 ... x 2 : concurrent access on AR
'For ADMIN <Semlock> and <Semadmin>' are set
' to Administration database
'For XXXX (database typed) <Semlock> is set
' to Development database
' and <Semadmin> is set to Administration database

Call Msg_Log (Array("1022" , "PTU908"))
'-----
WshEnv("PAC7IN") = Rep_SAVE & "\PG20"
WshEnv("PAC7OU") = Rep_SAVE & "\OLDPG"
Return = WshShell.Run("BVPTU908.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU908")

Call Msg_Log (Array("1022" , "PTU921"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU921", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PTU921", "PAC7GZ", Rep_TMP & "\WGZ.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU921", "PAC7PG", Rep_SAVE & "\OLDPG")
Call BvpEnv("PTU921", "PAC7ET", Rep_USR & "\PG20ET921.txt")
Return = WshShell.Run("BVPTU921.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU921")

'for the ADMIN base(<Z> space) :

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"

```

```

Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVP900.EXE" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025", _
                    "PAF900/ADMinistration", Return))
End If

If Return = 0 then

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
WshEnv("PAC7IE") = Rep_AUSR & "\PG20IE1A15.txt"
WshEnv("PAC7IF") = Rep_AUSR & "\PG20IF1A15.txt"
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025", _
                    "PACA15/ADMinistration", Return))
End If

End If

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WGZ.tmp")
Call DelFile (Rep_TMP & "\WME.tmp")
Call DelFile (Rep_TMP & "\WMV.tmp")
Call DelFile (Rep_TMP & "\WMW.tmp")
Call DelFile (Rep_TMP & "\WMX.tmp")
Call DelFile (Rep_TMP & "\WMY.tmp")
Return = 0
' To continue

'For the DEV base(<Y> space) :

```

```

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900", "PAC7MW", Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900", "PAC7MY", Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVP900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15", "PAC7IE", Rep_USR & "\PG20IE2A15.txt")
Call BvpEnv("PACA15", "PAC7IF", Rep_USR & "\PG20IF2A15.txt")
Call BvpEnv("PACA15", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15", "PAC7MV", Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

---

## Generation-Print Commands Retrieval (PG25)

### PG25 - Introduction

#### PRINCIPLE

The PG25 procedure retrieves the 2.5 release PG file, sequential image of the generation-print commands, in the 3.0 release new format.

It updates the development Database with the generation-print commands and the Administration Database with the 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.

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

### 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
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. : PGnnET921	Report	Error report ( nn= 20 ou 25 , version number)

### TRANSACTIONS FORMATTING: PAF900

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



Code	Physical name	Type	Label
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	Working 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 DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database Data file
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. Base - Base dir.: AN	Input	Administration Database Index file
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data file
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7DC	nul	Input	DSMS file of Development Database Elements
PAC7ME	Tmp dir.: WME	Input	Working file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	nul	Output	UPDT erroneous transactions (length=80)
PAC7RY	nul	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)

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

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

Return codes:

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

#### TRANSACTION FORMATTING: PAF900

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

UPDATE OF THE ADMINISTRATION DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	Admin. Base - Base dir.: AR	Output	Administration Database Data file
PAC7AN	Admin. Base - Base dir.: AN	Output	Administration Database Index file
PAC7AY	Admin. Base - Base dir.: AY	Output	Administration Database extension
PAC7AJ	Admin. Base - Base dir.: AJ	Output	Administration Database journal
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database Index file
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data file
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7DC	Base dir.: DC	Input	DSMS file of Development Database elements
PAC7ME	Tmp dir.: WME	Input	Work file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	nul	Output	UPDT erroneous transactions (length=80)
PAC7RY	nul	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

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

Return codes:

- 0 : OK without error
- 2 : warning error

- 4 : fatal error

## PG25 - Execution Script

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PG FILE SINCE 2.5
REM *
REM * -----
REM *
<job id=PG25>

<script language="VBScript">
Dim MyProc
MyProc = "PG25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

'-PG- Retrieval for the Aministration database
'and the Development database
'THUS the same steps are executed is this procedure :
' = PTUBAS + ... PACA15 ... x 2 : concurrent access on AR
'For ADMIN <Semlock> and <Semadmin>' are set to
' Administration database
'For XXXX (database typed) <Semlock> is set to
'Development database
' and <Semadmin> is set to Administration database

Call Msg_Log (Array("1022" , "PTU921"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU921", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PTU921", "PAC7GZ", Rep_TMP & "\WGZ.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU921", "PAC7PG", Rep_SAVE & "\OLDPG")
Call BvpEnv("PTU921", "PAC7ET", Rep_USR & "\PG25ET921.txt")
Return = WshShell.Run("BVPTU921.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU921")

'for the ADMIN base(<Z> space) :

Call Msg_Log (Array("1022" , "PAF900"))

```

```

'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025","PAF900/ADMinistration", Return))
End If

If Return = 0 then
Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG25IE1A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG25IF1A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
If Return <> 0 then
Call Msg_Log (Array("1025", _
"PACA15/ADMinistration", Return))
End If

End If

Call Msg_Log (Array("1024"))
'-----
Call DelFile (Rep_TMP & "\WGZ.tmp")
Call DelFile (Rep_TMP & "\WME.tmp")
Call DelFile (Rep_TMP & "\WMV.tmp")
Call DelFile (Rep_TMP & "\WMW.tmp")
Call DelFile (Rep_TMP & "\WMX.tmp")
Call DelFile (Rep_TMP & "\WMY.tmp")
Return = 0
' To continue

```

'For the DEV base(<Y> space) :

```
Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900", "PAC7GY", Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900", "PAC7MV", Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900", "PAC7MW", Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900", "PAC7MX", Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900", "PAC7MY", Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15", "PAC7IE", Rep_USR & "\PG25IE2A15.txt")
Call BvpEnv("PACA15", "PAC7IF", Rep_USR & "\PG25IF2A15.txt")
Call BvpEnv("PACA15", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15", "PAC7MV", Rep_TMP & "\WMV.tmp")
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>
```

---

## PEI Retrieval (PP25)

### PP25 - Introduction

#### PRINCIPLE

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

#### EXECUTION CONDITIONS

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

#### PRINTED OUTPUT

This procedure produces a report indicating the error encountered.

#### RESULT

The procedure generates a transaction file which contains the existing production environments, the list of the generated entities, the default environments (-GO of the Library), the list of production sessions in the 3.0 format, and updates the 3.0 development database.

#### NOTE

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

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

### PP25 - Input / Processing / Results

A `'*/`line with a user code and a password

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

The structure of the line is presented as follows:

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

## PP25 - Description of Steps

### MANAGEMENT OF PRODUCTION ENVIRONMENT: PTU923

Code	Physical name	Type	Label
PAC7PP	Save dir. : OLDPP	Input	Back up of production environment (old release) (old release)
PAC7AE	System - Skel. dir. : AE	Input	Error message
PAC7MB	User input	Input	User input
PAC7GY	Tmp dir. : WGY	Output	Records of production environments (length=310)
PAC7ET	User dir. : PP25ET923	Report	Report in case of error

### TRANSACTIONS FORMATTING: PAF900

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Input	Development Database data
PAC7AN	Base dir.: AN	Input	Development Database index
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp. dir.: WGZ	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	Working 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 DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database Data file
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. Base - Base dir.: AN	Input	Administration Database Index file
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data file
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7DC	nul	Input	DSMS file of Development Database Elements
PAC7ME	Tmp dir.: WME	Input	Working file
PAC7MV	Tmp dir.: WMV	Input	Update transactions
PAC7RB	nul	Output	UPDT erroneous transactions (length=80)
PAC7RY	nul	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

**PP25 - Execution Script**

```
REM * -----
REM *     VISUALAGE PACBASE
REM *
```

```

REM * -----
REM *          RETRIEVAL OF PP FILE
REM *
REM * -----
REM *
<job id=PP25>

<script language="VBScript">
Dim MyProc
MyProc = "PP25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU923"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU923","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU923","PAC7PP",Rep_SAVE & "\OLDPP")
Call BvpEnv("PTU923","PAC7ET",Rep_USR & "\PP25ET923.txt")
Return = WshShell.Run("BVPTU923.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU923")

Call Msg_Log (Array("1022" , "PAF900"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"

```

```

WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15", "PAC7IE", Rep_USR & "\PP25IE1A15.txt")
Call BvpEnv("PACA15", "PAC7IF", Rep_USR & "\PP25IF1A15.txt")
Call BvpEnv("PACA15", "PAC7ME", Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15", "PAC7MV", Rep_TMP & "\WMV.tmp")
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

---

## 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.: UV25	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. Base - Base dir.: AR	Input	Administration Database data
PAC7AN	Admin. Base - Base dir.: AN	Input	Administration Database index
PAC7AE	System - Skel. dir.: AE	Input	Error labels
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database data
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database index
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7GY	Tmp dir.: WGY	Input	Update transactions

Code	Physical name	Type	Label
PAC7MV	Tmp dir.: WMV	Output	Transactions formatting (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	Working file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Working file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Working file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Working file (length=743)

#### UPDATE OF THE ADMINISTRATION DATABASE: PACA15

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

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

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

Return codes:

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

## UV25 - Execution Script

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF UV FILE
REM *
REM * -----
REM *
<job id=UV25>

<script language="VBScript">
Dim MyProc
MyProc = "UV25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU922"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU922","PAC7ET",Rep_USR & "\UV25ET922.txt")
Call BvpEnv("PTU922","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU922","PAC7UV",Rep_BASE & "\UV25")
Return = WshShell.Run("BVPTU922.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU922")

Call Msg_Log (Array("1022" , "PAF900"))

```

```

'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Return = WshShell.Run("BVP900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\UV25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\UV25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

---

## Retrieval of MB Transactions (MB25)

### MB25 - Introduction

#### PRINCIPLE

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

#### REQUISITES

None.

#### PRINTED OUTPUT

This procedure prints a report of the errors encountered.

#### RESULT

This procedure generates a transaction file for the 3.0 UPDT procedure and a revoked transactions file.

### MB25 - Description of Steps

#### MB FILE RETRIEVAL: PTU926

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7AR	Base dir.: AR	Input	Development database data
PACGGR	Admin. Base - Base dir.: AR	Input	Administration database data
PACGGN	Admin. Base - Base dir.: AN	Input	Administration database index
PACGGU	Admin. Base - Base dir.: GU	Input	Administration database users
PAC7MB	User input	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

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF MB FILE  
REM *
```



```

REM * -----
REM *
<job id=MB25>

<script language="VBScript">
Dim MyProc
MyProc = "MB25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU926"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU926","PAC7MB",RepT_USR & "\MB25")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU926","PAC7EF",Rep_USR & "\MB25EF926.txt")
Call BvpEnv("PTU926","PAC7DD",Rep_USR & "\MB25DD926.txt")
Call BvpEnv("PTU926","PAC7ME",Rep_USR & "\MB25ME.txt")
Call BvpEnv("PTU926","PAC7MV",Rep_USR & "\MB25MV.txt")
Return = WshShell.Run("BVPTU926.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU926")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

---

## Retrieval of GY Transactions (GY25)

### GY25 - Introduction

#### PRINCIPLE

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

#### REQUISITES

None.

## PRINTED OUTPUT

This procedure prints a report of the errors encountered.

## RESULT

This procedure generates a transaction file for the 3.0 UPDP procedure and a revoked transactions file.

### **GY25 - Description of Steps**

#### GY FILE RETRIEVAL: PTU927

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7AR	Base dir.: AR	Input	Development database data
PACGGR	Admin. Base - Base dir.: AR	Input	Administration database data
PACGGN	Admin. Base - Base dir.: AN	Input	Administration database index
PACGGU	Admin. Base - Base dir.: GU	Input	Administration database users
PAC7GY	User input	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**

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF GY FILE  
REM *  
REM * -----  
REM *  
<job id=GY25>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "GY25"  
</script>
```

```

<script language="VBScript" src="INIT.vbs"/>
<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU927"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU927","PAC7GY",RepT_USR & "\GY25")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU927","PAC7EF",Rep_USR & "\GY25EF927.txt")
Call BvpEnv("PTU927","PAC7DD",Rep_USR & "\GY25DD927.txt")
Call BvpEnv("PTU927","PAC7ME",Rep_USR & "\GY25ME.txt")
Call BvpEnv("PTU927","PAC7MV",Rep_USR & "\GY25MV.txt")
Return = WshShell.Run("BVP7U927.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU927")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

---

## Retrieval of MB Transactions (MB30)

### MB30 - Introduction

#### PRINCIPLE

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

#### REQUISITES

None.

#### PRINTED OUTPUT

This procedure prints a report of the errors encountered.

## RESULT

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

### **MB30 - Description of Steps**

#### MB FILE RETRIEVAL: PTU928

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AR	Base dir. : AR	Input	Development database data
PACGGR	Admin Base - Base dir. : AR	Input	Administration database data
PACGGN	Admin Base - Base dir. : AN	Input	Administration database index
PACGGU	Admin Base - Base dir. : GU	Input	Administration database users
PAC7MB	User input	Input	3.0 MB transactions
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**

```
REM * -----  
REM *      VISUALAGE PACBASE  
REM *  
REM * -----  
REM *      RETRIEVAL OF MB FILE  
REM *  
REM * -----  
REM *  
<job id=MB30>  
  
<script language="VBScript">  
Dim MyProc  
MyProc = "MB30"  
</script>  
  
<script language="VBScript" src="INIT.vbs"/>  
  
<script language="VBScript">  
  
If c_error = 1 then Wscript.Quit (1) End If  
  
Call Msg_Log (Array("1022" , "PTU928"))  
'-----  
WshEnv("PAC7AE") = Rep_SKEL & "\AE"  
WshEnv("PAC7AR") = Rep_BASE & "\AR"
```

```

Call BvpEnv("PTU928","PAC7MB",RepT_USR & "\MB30")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU928","PAC7EF",Rep_USR & "\MB30EF928.txt")
Call BvpEnv("PTU928","PAC7DD",Rep_USR & "\MB30DD928.txt")
Call BvpEnv("PTU928","PAC7ME",Rep_USR & "\MB30ME.txt")
Call BvpEnv("PTU928","PAC7MV",Rep_USR & "\MB30MV.txt")
Return = WshShell.Run("BVPTU928.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU928")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

</script>
</job>

```

---

## Retrieval of GY Transactions (GY30)

### GY30 - Introduction

#### PRINCIPLE

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

#### REQUISITES

None.

#### PRINTED OUTPUT

This procedure prints a report of the errors encountered.

#### RESULT

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

## GY30 - Description of Steps

### GY FILE RETRIEVAL: PTU929

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AR	Base dir. : AR	Input	Development database data
PACGGR	Admin Base - Base dir. : AR	Input	Administration database data
PACGGN	Admin Base - Base dir. : AN	Input	Administration database index
PACGGU	Admin Base - Base dir. : GU	Input	Administration database users
PAC7GY	User input	Input	3.0 GY transactions
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

```
REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF GY FILE
REM *
REM * -----
REM *
<job id=GY30>

<script language="VBScript">
Dim MyProc
MyProc = "GY30"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU929"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU929", "PAC7GY", RepT_USR & "\GY30")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU929", "PAC7EF", Rep_USR & "\GY30EF929.txt")
```

```
Call BvpEnv("PTU929","PAC7DD",Rep_USR & "\GY30DD929.txt")
Call BvpEnv("PTU929","PAC7ME",Rep_USR & "\GY30ME.txt")
Call BvpEnv("PTU929","PAC7MV",Rep_USR & "\GY30MV.txt")
Return = WshShell.Run("BVPTU929.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU929")
```

```
Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)
```

```
Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)
```

```
</script>
</job>
```

---

## Retrieval of journal file PJ (PJ25)

### PJ25 - Introduction

#### PRINCIPLE

This procedure (PJ25) is used to change the PJ file, which is a journal file (transactions sequential file), into a new archival file in a 3.0 format.

#### PREREQUISITES

None.

#### PRINTED OUTPUT

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

#### RESULT

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

#### COMMENTS

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

This retrieval procedure must be used only for conversion from a 2.0 or 2.5 version into a 3.0 version.

Sometimes, to retrieve some transactions, it is necessary to search for information in a 3.0 database. But it is possible that such information no longer exist in the new database (example: session or library deleted). In such a case, the old transaction is retrieved with its 2.n format.

## PJ25 - Description of steps

### PROCESSING OF PJ TRANSACTIONS SEQUENTIAL FILE: PTU918

Code	Physical name	Type	Label
PAC7PJ	Save dir.: PJ25	Input	journal file old version
PAC7AE	System - skel. dir.: AE	Input	Error messages
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
PAC7JP	Save dir.: PJ	Sortie	Journal file in a 3.0 format (length=170)
PAC7ET	User dir.: PJ25ET918	Report	Report

## PJ25 - Execution JCL

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      RETRIEVAL OF PJ FILE
REM *
REM * -----
REM *
<job id=PJ25>

<script language="VBScript">
Dim MyProc
MyProc = "PJ25"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "PTU918"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AN") = Rep_BASE & "\AN"

```



```

WshEnv("PAC7AY") = Rep_BASE & "\AY"
Call BvpEnv("PTU918","PAC7PJ",Rep_SAVE & "\PJ25")
Call BvpEnv("PTU918","PAC7EF",Rep_USR & "\PJ25EF918.txt")
Call BvpEnv("PTU918","PAC7JP",Rep_SAVE & "\PJ")
Return = WshShell.Run("BVPTU918.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU918")

```

```

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

```

```

Call Msg_Log (Array("1023"))
'-----
Wscript.Quit (Return)

```

```

</script>
</job>

```

---

## Procedures - Summary Table of Changes

### LIST OF NEW PROCEDURES

Procedure	Comments
ARCH	Archiving Administration Database
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
REOR	Re-organization of Administration Database
REST	Restoration of Administration Database
PACS	Backup of Administration Database
VINS	Update of Administration Model
PACS	Management of development Database
GY25	Retrieval of GY file for UPDP
MB25	Retrieval of MB file for UPDT
GY30	Retrieval of GY file for UPDP PDP to 2.5
MB30	Retrieval of MB file for UPDT PDT to 2.5
PC25	Retrieval of development Database
PE25	Retrieval of user parameters (PE)
PG20	Retrieval of 2.0 generation-print commands (PG)
PG25	Retrieval of 2.5 generation-print commands (PG)
PJ25	Retrieval of journal 2.5

<b>Procedure</b>	<b>Comments</b>
UV25	Retrieval of PacTransfer parameters (UV)
PP25	Retrieval of PEI environment (PP)
CHPM	Environment and Database elements check report
CPPM	Comparison between Database and user configuration
EXPM	Extraction of environments
HIPM	Implementation of elements
SIPM	Generation simulation
TRED	PacTransfer: print of parameters

#### LIST OF PROCEDURES SUPPRESSED SINCE 2.5 RELEASE

<b>Procedure</b>	<b>Program</b>	<b>Comments</b>
CPSN	PTU850 PTU855	Integrated in PACX procedure
EMSN	PTU810	
MESN	PTU815	
MLIB	PTU100 PTU120	Integrated in PACS procedure
SASN	PTU130 PTU140	Integrated in PACS procedure
SAVE	PTU500	Integrated in PACS procedure
UXSR	UTIXSR	Integrated in PACS procedure
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACTI1	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	

<b>Procedure</b>	<b>Program</b>	<b>Comments</b>
RVKE	PREI40 PREI50	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	
VPU2	PVA320	
LVBL	PTULVB	
QREO	PTUN00 PTUN10 PTUN40	
RPPG	PTU908	
RPTD	PTAR20	



---

## Chapter 8. Components

---

### Server Environment Components

#### Introduction

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

- 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 skeleton generation file (SC file) used by the Batch generator function.
- A skeleton generation file (SG file) used by the On-Line Systems Development and Database generator functions.
- A skeleton generation file (SN file) used by the eBusiness generator function.
- A skeleton generation file (SR file), used by the Reverse generator function.
- A skeleton generation 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 skeleton generation 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.

### **Administration Database Backup**

The administration Database backup consists of two sequential generation files.

- Backup of the Database (PE).

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

- Backup of the journal (PJ).

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

## **Development Database**

### **Development Database Files**

The VA Pac Database files contain all data related to applications development.

- 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 option taken into account during restoration, the Database backup is either made of two sequential generation files (PC and PJ) or of four sequential generation files (PC, PD, PY and PJ).

- Database or Data backup (PC).

This is a sequential backup file of the Development Database components (Data (AR), Index (AN), and extension (AY))

- The backup of the Development Database index (PD).
- The backup of the Development Database extension 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 analysed (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.

The description of these files is the following one:

PQC FUNCTION

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

**- SPECIFIC MEMBERS OF PACBENCH QUALITY CONTROL FUNCTION**

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



---

## Chapter 9. Appendix

---

### Installation of the Administration Database Model

#### VINS - Introduction

The VINS procedure performs the batch update of the Administration Database using transactions provided by IBM.

#### EXECUTION CONDITIONS

The database must be closed to on-line processing.

#### ABNORMAL EXECUTIONS

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

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

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

#### VINS - Input / Processing / Results

The VINS procedure requires two types of user input:

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

The structure of the line is the following:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Library code
29	4	'VINS'	
33	1	'I'	Installation of IBM Meta-Entities

## PRINTED OUTPUT

The procedure outputs:

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

## RESULT

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

## NOTE

Extracted transactions to be used by the REOR procedure must be copied in a catalogued file by taking the comment line into account.

## **VINS - Description of Steps**

### Update of the Administration Database : VINS

<b>Code</b>	<b>Physical name</b>	<b>Type</b>	<b>Label</b>
PAC7AE	System - skel. dir.: AE	Input	error labels
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database Index
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database Users
PAC7AJ	Journal directory: AJ	Input	Administration Database Journal
PAC7AN	Admin. Base - Base dir.: AN	Input	Administration Database Index
PAC7AR	Admin. Base - Base dir.: AR	Input	Administration Database Data
PAC7AY	Admin. Base - Base dir.: AY	Input	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

Code	Physical name	Type	Label
PAC7WD	Tmp dir.: WWD	Input/Output	Extracted Transactions
PAC7ES	Tmp dir.: WES	Input/Output	Extracted Transactions
PAC7TD	Tmp dir.: WTD	Input/Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR on Administration Base
PAC7MX	null	Output	Non extracted entities
PAC7IA	User dir.: VINGIA	Report	General Report of Programs sequence
PAC7EE	User dir.: VINGEE	Report	Report
PAC7EQ	User dir.: VINGEQ	Report	Report
PAC7EU	User dir.: VINGEU	Report	Report
PAC7ER	User dir.: VINGER	Report	Report
PAC7EZ	User dir.: VINGEZ	Report	Report
PAC7DD	User dir.: VINGDD	Report	Report
PAC7IE	User dir.: VINGIE	Report	Report
PAC7IF	User dir.: VINGIF	Report	Report
PAC7IG	User dir.: VINGIG	Report	Report
PAC7IH	User dir.: VINGIH	Report	Report

Return codes:

- 0: No error
- 4: Warning
- 8: No access authorization for batch procedure or invalid database (in such a case, restart the procedure with 'D' in column 16)
- 12: Input-output error on a file.

## VINS - Execution Script

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      - DICTIONARY UPDATING WITH IBM MODEL DEVPT -
REM *
REM * -----
REM *
REM * THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
REM * DATABASE, BASED ON TRANSACTIONS PROVIDED.
REM *
REM * INPUT :
REM * - USER IDENTIFICATION LINE (REQUIRED)

```

```

REM *      COL 2 : '*'
REM *      COL 3 : USERIDXX
REM *      COL 11 : PASSWORD
REM *      COL 27 : ' ' - NO UPDATE
REM *          'S' - UPDATE SIMULATION WITH PRINTING OF
REM *                  OF LIST OF U.E.'S TO BE CANCELLED
REM *          'F' - FORCING THE CANCELLATION OF U.E.'S
REM *                  WITH THE SAME CODES IN LOWER
REM *                  LEVEL LIBRARIES
REM * -----
REM *
<job id=VINS>

```

```

<script language="VBScript">
Dim MyProc
MyProc = "VINS"
</script>

```

```

<script language="VBScript" src="INIT.vbs"/>

```

```

<script language="VBScript">

```

```

If c_error = 1 then Wscript.Quit (1) End If

```

```

Call Msg_Log (Array("1029" ))
'-----
Call StateList (base, statusL)

```

```

If c_error = 1 then Wscript.Quit (1) End If

```

```

Call Msg_Log (Array("1022" , "VINS"))
'-----

```

```

WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("VINS", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("VINS", "PAC7ES", Rep_TMP & "\WES.tmp")
Call BvpEnv("VINS", "PAC7DD", Rep_USR & "\VINSDD.txt")
WshEnv("PAC7EE") = "NUL"
Call BvpEnv("VINS", "PAC7EQ", Rep_USR & "\VINSEQ.txt")
Call BvpEnv("VINS", "PAC7ER", Rep_USR & "\VINSEI.txt")
Call BvpEnv("VINS", "PAC7EU", Rep_USR & "\VINSEU.txt")
Call BvpEnv("VINS", "PAC7EZ", Rep_USR & "\VINSEZ.txt")
Call BvpEnv("VINS", "PAC7IA", Rep_USR & "\VINSIA.txt")
Call BvpEnv("VINS", "PAC7IE", Rep_USR & "\VINSIE.txt")
Call BvpEnv("VINS", "PAC7IF", Rep_USR & "\VINSIF.txt")
Call BvpEnv("VINS", "PAC7IG", Rep_USR & "\VINSIG.txt")
Call BvpEnv("VINS", "PAC7IH", Rep_USR & "\VINSIH.txt")

```

```

If base = "ADMIN" Then
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETAD"
Else
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETBA"
End If

'Example of Input File extracted from PACX/CPSN :
' Call BvpEnv("VINS","PAC7xx",RepT_USR & "\PACXxx.txt")
'With RepT_USR is Global User Directory.
Call BvpEnv("VINS","PAC7TD",Rep_USR & "\TD.txt")

WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("VINS","PAC7MR",Rep_TMP & "\WMR.tmp")
WshEnv("PAC7MX") = "NUL"
Call BvpEnv("VINS","PAC7WD",Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS","SYSEXT",Rep_TMP & "\VINSSYS.tmp")
Call RunCmdLog ("BVPVINS")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "VINS")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
WshVolEnv("RC") = Return
Wscript.Quit (Return)

</script>
</job>

```

---

## Installation of the Development Database Model

### VINS - Introduction

The VINS procedure performs the batch update of the network using transactions provided by IBM.

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

### EXECUTION CONDITIONS

The database must be closed to on-line processing.

### ABNORMAL EXECUTIONS

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

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

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

## VINS - Input / Processing / Results

The VINS procedure requires two types of user input:

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

The structure of the line is the following:

Position	Length	Value	Meaning
2	1	'*'	Line code
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	'***'	Library code
29	4	'VINS'	
33	1	'I'	Installation of IBM Meta-Entities
		'R'	Retrieval of User Entities with the 'extension' format
		' '	'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 network is ready for either on-line or batch use.

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

NOTE

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

**VINS - Description of Steps**

UPDATE OF THE NETWORK: VINS

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error Labels
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database Index
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database Users
PAC7AJ	Journal dir.: AJ	Input	Development Database Journal
PAC7AN	Base dir.: AN	Input	Development Database Index
PAC7AR	Base dir.: AR	Input	Development Database Data
PAC7AY	Base dir.: AY	Input	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	Tmp dir.: WTD	Input/Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR
PAC7MX	null	Output	Non extracted entities
PAC7IA	User dir.: VINSIA	Report	Complete printing of programs sequence

Code	Physical name	Type	Label
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 : Erroneous record in journal file (Date or session number not numeric)
- 8: No access authorization for batch procedure or invalid database (in such a case, restart the procedure with 'D' in column 16)
- 12: Input-output error on a file

## VINS - Execution Script

```

REM * -----
REM *      VISUALAGE PACBASE
REM *
REM * -----
REM *      - DICTIONARY UPDATING WITH IBM MODEL DEVPT -
REM *
REM * -----
REM *
REM * THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
REM * DATABASE, BASED ON TRANSACTIONS PROVIDED.
REM *
REM * INPUT :
REM * - USER IDENTIFICATION LINE (REQUIRED)
REM *   COL 2 : '*'
REM *   COL 3 : USERIDXX
REM *   COL 11 : PASSWORD
REM *   COL 27 : ' ' - NO UPDATE
REM *           'S' - UPDATE SIMULATION WITH PRINTING OF
REM *                OF LIST OF U.E.'S TO BE CANCELLED
REM *           'F' - FORCING THE CANCELLATION OF U.E.'S
REM *                WITH THE SAME CODES IN LOWER
REM *                LEVEL LIBRARIES
REM * -----
REM *
<job id=VINS>

```



```

<script language="VBScript">
Dim MyProc
MyProc = "VINS"
</script>

<script language="VBScript" src="INIT.vbs"/>

<script language="VBScript">

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1029" ))
'-----
Call Statelist (base, statusL)

If c_error = 1 then Wscript.Quit (1) End If

Call Msg_Log (Array("1022" , "VINS"))
'-----
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("VINS", "PAC7BM", Rep_TMP & "\WBM.tmp")
Call BvpEnv("VINS", "PAC7ES", Rep_TMP & "\WES.tmp")
Call BvpEnv("VINS", "PAC7DD", Rep_USR & "\VINSDD.txt")
WshEnv("PAC7EE") = "NUL"
Call BvpEnv("VINS", "PAC7EQ", Rep_USR & "\VINSEQ.txt")
Call BvpEnv("VINS", "PAC7ER", Rep_USR & "\VINSER.txt")
Call BvpEnv("VINS", "PAC7EU", Rep_USR & "\VINSEU.txt")
Call BvpEnv("VINS", "PAC7EZ", Rep_USR & "\VINSEZ.txt")
Call BvpEnv("VINS", "PAC7IA", Rep_USR & "\VINSIA.txt")
Call BvpEnv("VINS", "PAC7IE", Rep_USR & "\VINSIE.txt")
Call BvpEnv("VINS", "PAC7IF", Rep_USR & "\VINSIF.txt")
Call BvpEnv("VINS", "PAC7IG", Rep_USR & "\VINSIG.txt")
Call BvpEnv("VINS", "PAC7IH", Rep_USR & "\VINSIH.txt")

If base = "ADMIN" Then
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETAD"
Else
WshEnv("PAC7MA") = Rep_SKEL & "\BVPMETBA"
End If

'Example of Input File extracted from PACX/CPSN :
' Call BvpEnv("VINS", "PAC7xx", RepT_USR & "\PAC7xx.txt")
'With RepT_USR is Global User Directory.
Call BvpEnv("VINS", "PAC7TD", Rep_USR & "\TD.txt")

```

```

WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("VINS","PAC7MR",Rep_TMP & "\WMR.tmp")
WshEnv("PAC7MX") = "NUL"
Call BvpEnv("VINS","PAC7WD",Rep_TMP & "\WWD.tmp")
Call BvpEnv("VINS","SYSEXT",Rep_TMP & "\VINSSYS.tmp")
Call RunCmdLog ("BVPVINS")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "VINS")

Call Msg_Log (Array("1024"))
'-----
Call DeleteFldr (Rep_TMP)

Call Msg_Log (Array("1023"))
'-----
WshVolEnv("RC") = Return
Wscript.Quit (Return)

</script>
</job>

```





Part Number: DELNT001304A - 6038

Printed in U.S.A.

(1P) P/N: DELNT001304A - 6038

