VisualAge Pacbase

Installation Guide
BULL GCOS7 Server & Client Components

Version 3.0

VisualAge Pacbase

Installation Guide
BULL GCOS7 Server & Client 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.

First Edition (January 2003)

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 1983,2003. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1983,2003

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.

© Copyright IBM Corp. 1983,2003

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.

© Copyright IBM Corp. 1983,2003

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,

© Copyright IBM Corp. 1983,2003

- one or more development Database(s).

THE CLIENT ENVIRONMENT

It consists of the following components:

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

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

Contents of Supply

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

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

Bibliography

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

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

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

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

Chapter 2. Prerequisites

Prerequisites for Server Environment

Hardware and Software

Processor: GCOS-7

The system running the product must have the following characteristics:

- Operating system : GCOS-7 V9 or higher
  - TDS level : GCOS-7 V9 or higher
  - Supported messages : VIP7700, QUESTAR, VIP7800, IBM3270

Installation media : cartridge 3480

Monitor : TDS/GCOS7

The data file (AR), index file (AN), journal file (AJ), extension file (AY) and error message file (AE) are updated on-line. Therefore, they must be protected by the TDS journalization option (Before Journal).

You should anticipate 36000K for back storage.

COBOL: VA PAC programs have been compiled by a compiler version V14.

GENERAL INFORMATION - HOW THE SYSTEM RUNS

The general characteristics are:

- The transaction code value is set by the user. The fourth character indicates the type of terminal to use:

  '1' : only QUESTAR screens may be used

  '2' : only IBM 3270 screens may be used

  '3' : only VIP 7800 screens may be used

  Other : all terminal types may be used, display mode is automatically managed when the user accesses the transaction.

  In order to benefit from VIP screens which are limited to 128 input fields, the transaction code must be followed by '/128'. This regroups several variable fields. The user may enter this option in the choice field of any VA Pac screen.

  Example: regardless of screen type

© Copyright IBM Corp. 1983,2003

PB00 ----> 192 fields mode

PB00 /128 ----> 128 fields mode (VIP or QUESTAR)

- Updates are serialized, i.e. VA Pac system manages simultaneous access by queuing update TPRs.
- 'FT' entered in the operation field on the PACBASE initial screen ensures a correct exit. The following message is displayed: 'END OF CONVERSATION'.
- The names of Sharable modules used by Va Pac (TPR,... ,TPR6) can no be modified. Many other programs are linked to TDS in generation phase.

Disk Space

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

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

Disk space for the installation	Total	UFAS	Non UFAS
Total installation	200	110	90
Sub-total System	140	75	65
Sub-total user files for installation tests	60	35	25

Prerequisites for Client Environment

Hardware

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

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

Disk Space

Required disk space:

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

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

Communication

To enable the communication between the workstation components and the servers in a mainframe

environment, the communication protocol must be:

- TCP-IP Access TDS

The following text only indicates elements of specific configuration to use VisualAge Pacbase Client components.

Bull products required:

- on GCOS7
  - GCOS 7-V7 (TS7560 minimum), GCOS 7-V8 (TS8560 minimum) or GCOS 7-V9 (TS9662 minimum)
  - TDS-TCP/IP
  - SOCKG7 V4.1.0
  - GCOS 7 OPEN 7 release V5 or higher
- Windows client
  - TDS-TCP/IP API for Workstation
- AIX client (4.2.1 or higher)
  - TDS-TCP/IP API for Unix

    The TDS must be regenerated with the TCP-IP option.

Example of location in vaplocat.ini :

- TCP/IP Access TDS/GCOS7

```
vaplocat.ini
<TCP-IP_Access_TDS>
COMM_TYPE=TCPTDSDS>
IXO_HOSTNAME=BC0E
IXO_TDSNAME=VPD5
```

```
IXO_PROJECT=PT
IXO_BILLING=PACBASE0
IXO_TERMID=
IXO_TRANSID=VTCP
IXO_DATACONVERT=Y
COMMENT HOST_ENCODING=297
IXO_USERID=PTTD2
```

Chapter 3. Installation of Server Environment

Introduction

The installation procedure is executed in three main steps:

- Preparation for installation,
- Installation,
- On-line and batch tests.

A special installation tape is provided by CGI. The whole installation process is described in this chapter.

Before executing the actual installation, the user must be familiar with the technical characteristics of VA Pac described in this manual. This information is necessary to prepare the environment required for the installation procedure (disk space, definition of the TDS and its users on the catalog, etc.).

Installation tape

•

The installation tape contains the following files:

Rank	Label	Contents
1	SV.PROC	Procedures library
2	SVF.SC	BATCH-language skeleton file
3	SVE.SC	
4	SVF.SG	OLSD, DBD and DATA skeleton file
5	SVE.SG	
6	SVF.SR	COBOL generator skeleton file
7	SVE.SR	
8	SVF.SS	C/S generator skeleton file
9	SVE.SS	
10	SVF.SN	C/S generator skeleton file
11	SVE.SN	
12	SV.AE0	Error-message and documentation
13	SV.PC	Test development database
14	SV.PE	Test administration database

© Copyright IBM Corp. 1983,2003

Rank	Label	Contents
15	SV.CUB	Batch compile-units
16	SV.CUT	TPR Compile-units
17	SV.SP	PAF-PDM variable skeleton file
18	SV.SF	PAF-PDM fixed skeleton file
19	SV.METAD	Administration model
20	SV.METBA	Development database model
21	SV.ABOUT	Information on the release

Complete jcl installation

The installation is executed in four steps:

1. Creation of an auto-attachable catalog and two directories:

   BVP and $BVP . SYS

   where $BVP is the root directory of the VA Pacbase environment.
2. Allocation of a library for $BVP . SYS . PROC procedures, by BLIB (members=300, size=1 cylinder, compact).
3. Loading of the library, using the first file on the tape 'SV . PROC', by LMN:
4. Adaptation of the procedures to the site specific need. The adjustment is executed modifying in the 'ZZVALS' member the values chosen for the installation, and executing the following commands:

- S: LMN SL $BVP.SYS.PROC
- C: INF SV.PROC : cartridge : CT/xxxx
- C: MV INF : * BRIEF REPLACE
- C: INF
- C: EXEC ZZEXEC VL = ZZJCL BRIEF
- C: EJ ZZJCL ,, $BVP . SYS. PROC

NOTE :: The VA Pac TDS is usually linked in the VA Pac library of batch load-modules. For this reason, the value of $TDSNAME must be different from the values of VA Pacbase standard load-modules, in particular 'BVPACB' and 'BVPACQ' (See the list of Batch Load-Modules in the corresponding library).

1. Adaptation of the procedures to the site specific need. The adjustment is executed modifying in the 'ZZVALS' member the values chosen for the installation, and executing the following commands:

- C: EXEC ZZEXEC VL = ZZJCL BRIEF
- C: EJ ZZJCL ,, $BVP . SYS. PROC

NOTE :: The VA Pac TDS is usually linked in the VA Pac library of batch load-modules. For this reason, the value of $TDSNAME must be different from the values of VA Pacbase standard load-modules, in particular 'BVPACB' and 'BVPACQ' (See the list of Batch Load-Modules in the corresponding library).

```
COMM 'VisualAge Pacbase 3.0';
*****************************************
*        INSTALLATION PARAMETERS        *
*                                       *
*  REPLACE, IF NEEDED, THE DEFAULT      *
*  VALUE OF EACH PARAMETER.             *
*  EACH PARAMETER LINE IS FORMATTED AS  *
*  FOLLOWS:                             *
*    $NNNNN = VALUE                     *
*                                       *
*  THIS FILE IS PROCESSED BY AN EDITOR  *
*  PROGRAM WHICH CHANGES PARAMETER LINES *
*  INTO SUBSTITUTION COMMANDS.          *
*                                       *
*  SUBSEQUENTLY,                        *
*                                       *
*  - ALL LINES WHOSE FIRST NON-BLANK    *
*    CHARACTER IS NOT A DOLLAR SIGN     *
*    ARE CONSIDERED AS COMMENTS.        *
*                                       *
*  - THE EQUAL SIGN (DELIMITER) CANNOT  *
*    BE USED IN A PARAMETER VALUE.      *
*****************************************
******** VA PAC ROOT DIRECTORY          *
$BVP     =  PT.PB300
******** DEFAULT VA PAC USER NAME       *
$USER    =  TEST
******** VA PAC CATALOG NAME            *
$CATNAME =  PT
******** VA PAC TDS NAME                *
$TDSNAME =  VPD1
******** INSTALLATION MEDIA NAME        *
$TAPENAME=  PKJ000
******** INSTALLATION MEDIA TYPE        *
$TAPETYPE=  CT/36T
******** DEVELOPMENT DATABASE NAME      *
$BASE    =  PB00
******** LANGUAGE CODE E=ENGLISH F=FRENCH *
$LANG    =  F
******** REPORT INVOKED JCLS IN REPORTS *
$LIST    =  &LIST
******** BATCH FILES CI SIZE            *
$CISEQ   =  14336
******** VA PAC SYSTEM MEDIA TYPE       *
$DISCTYPE  =  MS/FSA
******** VA PAC SYSTEM MEDIA NAME       *
$DISCNAME  =  SDABAU
******** MEDIA TYPE FOR TEMPORARY FILES *
$DVTM      =  MS/FSA
```

```
******** MEDIA NAME FOR TEMPORARY FILES   *
$MDTM      =  SDABAU
******** PC BACKUP MEDIA TYPE             *
$PCMEDIA   =  DISC
******** PC BACKUP MEDIA NAME             *
$PCMDNAME  =  SDABAU
******** PD BACKUP MEDIA TYPE             *
$PDMEDIA   =  DISC
******** PD BACKUP MEDIA NAME             *
$PDMDNAME  =  SDABAU
******** PJ BACKUP MEDIA TYPE             *
$PJMEDIA   =  DISC
******** PJ BACKUP MEDIA NAME             *
$PJMDNAME  =  SDABAU
******** PY BACKUP MEDIA TYPE             *
$PYMEDIA   =  DISC
******** PY BACKUP MEDIA NAME             *
$PYMDNAME  =  SDABAU
******** PACTABLES ROOT DIRECTORY         *
$TRTAB     =  PT.PB300
******** DSMS FILE NAME (DUMMY)           *
$DSMS      =  DUMMY
```

The procedures are installed in the $BVP.SYS.PROC source directory. In the procedures, the &BASE parameter represents the database code during the execution.

An invoke JCL is used to run procedures, by the ej system command:

EJ IVPROC,,$BVP.SYS.PROC

The value of a parameter can be modified by entering a value in an invoke jcl. For this reason, it is recommended to copy all these invoke jcls in a $BVP.DATA.&BASE.INPUT library directly linked to the database in use.

It also can be duplicated for each user, who so can keep its own procedures input.

Installation process

Once the JCL is obtained, a VA Pac installation is executed in 8 main steps:

- System file allocation,
- VA Pac management TDS preparation,
- Unloading of files and programs,
- VA Pac management TDS generation,
- Link-edit of programs,
- Unloading of error labels files,
- Initialization of users management file,

- Creation of an administration database,
- Update of the administration metamodel,
- Creation of a VA Pac test Database,
- Update of the development metamodel,
- TDS submission.

System files allocation (INPRPB)

The system file allocation is executed by the 'INPRPB' member, which is included in $BVP.SYS.PROC.

Though several files are allocated by some of the following procedures, this step is useful to check that the disk space needed is available.

The 'INPRPB' member is a sequence of PREALLOC/LIBALLOC, which may be logically split as follows:

- Allocation of permanent UFAS files:
  - $BVP.SYS.SG
  - $BVP.SYS.SC
  - $BVP.SYS.SR
  - $BVP.SYS.AE0
  - $BVP.SYS.AE
    - $BVP.SYS.SF
    - $BVP.SYS.SN
    - $BVP.SYS.SP
    - $BVP.SYS.SS
    - $BVP.SYS.GS
    - $BVP.SYS.JQ
- Allocation of libraries:
  - $BVP.SYS.PGM
  - $BVP.SYS.CUB
  - $BVP.SYS.CUT
  - $BVP.SYS.CUP
  - $BVP.DATA.&BASE.INPUT
  - $BVP.DATA.&BASE.USERS
- Allocation of ADMIN database files:
  - $BVP.DATA.ADMIN.AJ
  - $BVP.DATA.ADMIN.AN
  - $BVP.DATA.ADMIN.AR
  - $BVP.DATA.ADMIN.AY

- $BVP.DATA.ADMIN.PC
- $BVP.DATA.ADMIN.PD
- $BVP.DATA.ADMIN.PJ
- $BVP.DATA.ADMIN.PY

- Allocation of VA Pac database files:
  - $BVP.DATA.&BASE.AJ
  - $BVP.DATA.&BASE.AN
  - $BVP.DATA.&BASE.AR
  - $BVP.DATA.&BASE.AY
  - $BVP.DATA.&BASE.PC
  - $BVP.DATA.&BASE.PD
  - $BVP.DATA.&BASE.PJ
  - $BVP.DATA.&BASE.PY
- Allocation of batch working files:

  WARNING!:

  If the PC, PD, PE, PJ sequential files are to be backed up on tape, the names of the backup tapes must be specified in the parameters of the JCL member ALxx.

TDS preparation (INMPRE)

The 'INMPRE' member included in the '$BVP.SYS.PROC' library starts the execution of the procedure.

WARNING: the cohabitation of the VA Pac installation and other applications in the same TDS is difficult. So the chosen TDS will be dedicated to VisualAge Pacbase only.

Loading of files & programs (INUNLD)

The installation of files and programs is executed by the 'UNLD' member which is a sequence of LIBMAINT and CREATE. This job contains 14 steps and may be long in executing time.

- Copy of jcls for submitting procedures in the library
  - $BVP.DATA.&BASE.INPUT
- Copy of programs in the libraries
  - $BVP.SYS.CUB
  - $BVP.SYS.CUT
- Copy of TDS sub-programs in the library
  - $BVP.SYS.CUP
- Copy of UFAS files in the libraries

- $BVP.SYS.SC
- $BVP.SYS.SG
- $BVP.SYS.SR
- $BVP.SYS.SP
- $BVP.SYS.SF
- $BVP.SYS.SS
- $BVP.SYS.SN
- $BVP.SYS.AE0

TDS generation (INMGEN)

The generation is executed by the 'INMGEN' procedure included in the $BVP.SYS.PROC library.

IMPORTANT: NOTE:

TDS general parameters (SIMULTANEITY, TERMINALS,...) should be checked, and modified if necessary, according to the site's environment and activity.

Programs link-edit (INLKT- & INLKB-)

The programs link-edit is executed by four JCLs for on-line programs and seven JCLs for batch programs:

- INLKTD : folders servers programs
- INLKTF : puf programs
- INLKTQ : transactional programs
- INLKTV : logical views servers programs
- INLKBG : print-generation programs
- INLKBP : paf programs
- INLKBQ : quality control programs
- INLKBU : batch update programs
- INLKBV : meta-entities update programs
- INLKBX : all-purpose extraction programs
- INLKBA : other batch programs

Loading of error messages & on-line help (IVINAE)

INAE procedure

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

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

English is the default language code.

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

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

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

Installation of the Administration database

Users codes initialization (IVINGU)

INGU procedure

This JCL should be submitted at the first installation only.

It defines, and then loads the GU file with the 'TEST' and 'ADMIN' user codes. These user codes will be used when the test JCL lines are run.

Loading of the Administration database (INRSAD)

Modify and submit the INRSAD jcl

This JCL must be submitted at the first installation only.

It runs the REST procedure in order to restore the administration Database.

NOTE

If you have a database that is older than the 3.0 version, consult the chapter dedicated to the retrieval of user parameters.

CAUTION

To use VisualAge Pacbase, you need an access key. You must enter here the key that you were given at delivery. It only allows to execute some procedures, especially the installation procedures.

It must be updated with the Administrator Workbench to take into account the whole product.

Installation of PCM archiving file (ALQJ)

ALQJ and ALJQ JCLs

These JCLs should be executed only at the first installation.

Loading of the Administration model (IVVING)

VING procedure

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

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

Installation of the Development database

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

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

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

Loading of the Test database (INREST)

Submit the INREST jcl without any modification

This job executes the REST procedure.

To install the test database you are provided with, run the JCL and do not add any change.

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

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

If the version of your database is previous to 3.0, see in the 'Database Retrieval' chapter, the sub-chapter dedicated to the retrieval of user parameters.

Loading of the Development model (IVVINS)

VINS procedure

This JCL must be submitted at the first installation and at each re-installation of Development Database.

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

Complement - Pac/Impact (IVINFP, ININFQ & IVIINI)

To use the impact analysis module (optional), it is necessary to create the necessary files:

- by the IINI procedure for the working files
- by the INFP procedure for the FP file
- by the INFQ procedure for FQ file

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

Chapter 4. Installation/Re-installation of Client Components

Things to Know Before Installing

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

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

  ```
  www.microsoft.com/msdownload/vbscript/scripting.asp
  ```
- Both Administrator & Developer workbench and eBusiness Tools components require, for their online help, that Netscape or Internet Explorer 5.5 be installed.
- The installation of a Client component does not require the prior installation on the server of the VA Pac Database(s) to which it will connect.

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

Root Directory

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

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

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

NOTE: Do not use blank characters in directory names.

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

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

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

© Copyright IBM Corp. 1983,2003

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

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

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

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

Installation Startup

Insert the installation CD.

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

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

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

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

Choose the Client component you want to install.

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

Fundamentals of VA Pac Client-Server Communication

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

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

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

COMMUNICATION PROTOCOL

- It is the TDS TCP/IP protocol. You must install and configure this protocol on the (or all) workstation(s) where the VisualAge Pacbase Middleware is installed.

  For more information on this configuration, see subchapter 'Additional Information', section 'How to configure the TDS TCP/IP protocol'.

THE MIDDLEWARE

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

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

However you can also choose a communication via a gateway.

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

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

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

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

COMMUNICATION FILES

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

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

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

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

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

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

THE VAPLOCAT.INI FILE

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

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

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

A location :

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

THE BASES.INI FILE

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

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

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

Administrator & Developer workbench

If the JDK (Java Developer's Kit) is not installed on your workstation, its installation will automatically take place.

For this installation, as for that of the Administrator & Developer workbench, the root being used depends on the current installation context. For complete details, refer to this chapter's first page.

The installation script then asks you to select the one or more elements to install:

- Administrator workbench

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

- Developer workbench

  Developer workbench includes the following modules:

  – Batch module,

  – eBusiness module and three eBusiness tools:

    - Proxy Generator,
    - Location Editor,
    - Services Test Facility.

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

  – Rational Rose Bridge.

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

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

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

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

  For complete details on this part of the installation, refer to the Middleware subchapter.

- If you choose the gateway option, enter the IP address of the gateway here. The installation of the middleware on the Developer workstation - also necessary in this context - will then start automatically after the installation of the workbench.

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

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

  To do this, a window enables you to enter:

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

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

2. The Database logical code.

   Maximum length: 4 characters.

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

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

3. The location name

   Maximum length: 20 characters.

   Default: Location-1

   More than one Database may be associated with one location.

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

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

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

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

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

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

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

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

START-UP FILES

The start-up files are :

- wb_admin.bat
- wb_batch.bat
- wb_eBusiness.bat

- wb_cfm.bat

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

The cfm module is a utility that allows you to inhibit the display of selected browsers unused by your site's teams and/or to provide for the display of browsers specific to Meta-Entities defined at your site. Consult the VisualAge Support team for more information.

START MENU / PROGRAMS CHOICE

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

```
Administrator-Developer workbench
                                  Administration
                                  Batch
                                  eBusiness
                                  cfm
```

Open Jade and Tidy / Publishing facility

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

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

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

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

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

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

eBusiness Tools

The eBusiness Tools are:

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

This installation allows the eBusiness Tools to be used independently of Developer workbench, without a connection to the VisualAge Pacbase server.

eBusiness tools are installed as VisualAge for Java features and tools. The specificity of VisualAge Pacbase Connector is to run in VisualAge for Java only.

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

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

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

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

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

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

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

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

START-UP FILES

The start-up files are :

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

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

```
ebusinesstools\bin
```

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

START MENU / PROGRAMS CHOICE

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

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

VisualAge Pacbase WorkStation

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

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

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

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

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

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

In the next panel, select the elements to install:

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

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

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

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

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

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

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

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

To do this, a panel enables you to enter:

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

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

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

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

   Complete details on these parameters are given at the end of this subchapter.
2. The Database logical code.

   Maximum length: 4 characters.

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

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

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

3. The location name

   Maximum length: 20 characters.

   Default: Location-1

   More than one Database may be associated with one location.

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

   You select the mode via two check boxes.

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

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

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

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

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

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

START-UP FILE

The start-up file is :

```
pexec.exe
```

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

START MENU / PROGRAMS CHOICE

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

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

INSTALLATION PARAMETERS FILE

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

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

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

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

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

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

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

DESCRIPTION OF THE PARAMETERS FILE

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

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

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

```
001 Station Version             [300F]
002 Server                      [PACBASE]
003 Communication Manager       [MWCOM]
004 Communication Parameters    [MWCOM]
005 System                      [WINDOWS]
006 Method                      [MER]
```

```
007 EXE disk                    [C]
008 EXE disk(default)           [C]
009 System Data Disk            [C]
010 User Data Disk              [C]
011 Connection execution mode   [E]
```

The Pacbase.dat file should not be destroyed.

The possible values for the Methodology parameter are:

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

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

Pacbase Web Connection

INSTALLATION PREREQUISITE

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

`http://www.perl.com`

To install the interpreter, follow the indications given below.

INSTALLATION

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

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

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

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

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

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

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

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

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

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

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

EXECUTION FILE

The Pacbase Web Connection execution file is:

```
Pacweb.exe
```

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

INSTALLATION OF THE CONTEXT SERVER AS AN NT SERVICE

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

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

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

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

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

When the computer boots up, it is automatically executed.

To uninstall the service, enter the following command:

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

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

Middleware

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

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

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

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

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

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

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

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

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

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

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

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

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

LIST OF PARAMETERS

- VAP SOCKET

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

Additional Information

How to Configure the TDS/GCOS7 TCP/IP Protocol

The following text only indicates elements of specific configuration to use VisualAge Pacbase Client components.

Bull products required:

- on GCOS7
  - GCOS 7-V7 (TS7560 minimum), GCOS 7-V8 (TS8560 minimum) or GCOS 7-V9 (TS9662 minimum)
  - TDS-TCP/IP
  - SOCKG7 V4.1.0
  - GCOS 7 OPEN 7 release V5 or higher
- Windows client
  - TDS-TCP/IP API for Workstation
- AIX client (4.2.1 or higher)
  - TDS-TCP/IP API for Unix

    The TDS must be regenerated with the TCP-IP option.

Example of location in vaplocat.ini :

- TCP/IP Access TDS/GCOS7

```
vaplocat.ini
<TCP-IP_Access_TDS>
COMM_TYPE=TCPTDSDS>
IXO_HOSTNAME=BC0E
IXO_TDSNAME=VPD5
IXO_PROJECT=PT
IXO_BILLING=PACBASE0
IXO_TERMID=
IXO_TRANSID=VTCP
IXO_DATACONVERT=Y
COMMENT HOST_ENCODING=297
IXO_USERID=PTTD2
```

Editing Communication Parameters

The bases.ini File

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

By default this file's access path is:

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

NOTE: All the parameters which may be present in the bases.ini file are not explained here. In fact, a number of these parameters allow finer

middleware settings, particularly used by proxies (generated by the eBusiness Tools). These parameters are used separately from the bases.ini file and are documented in the Proxy Programming Interface manual.

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

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

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

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

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

DESCRIPTION OF A SECTION'S CONTENTS

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

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

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

- VAPac Middleware: indicates that the user will have to give his/her code and password when he/she connects to the host and to the Database (default option).

- communicationAdapter = indicates the communication mode in use.
  - DIRECT: local middleware
  - GATEWAY: remote middleware (via the VisualAge Pacbase gateway)

The following parameters vary according to the chosen option.

PARAMETERS FOR DIRECT ADAPTER (LOCAL MIDDLEWARE)

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

  Default: ..\common\vaplocat.ini

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

- location = location name for the Database

  Maximum length: 20 characters.

  Default: Location-1

  More than one Database can point to the same location.

  REMINDER

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

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

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

- traceLevel = trace level of the middleware execution. Its possible values are:
  - 0 : no trace
  - 1 : trace with errors (default)
  - 2 : standard trace, not detailed
  - 3 : trace for information
  - 4 et + : trace for debug

- codePageFile = path and name of the file which contains the conversion table of the code pages.

  Default: ..\middleware\CharConv.txt

PARAMETERS FOR GATEWAY ADAPTER (REMOTE MIDDLEWARE)

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

  Default: 127.0.0.1 for a local host
- port = value of the IP port where the gateway receives the client requests.

  Default: 5647
- location = location name for the Database

  Maximum length: 20 characters.

  Default : Location-1

  More than one Database can point to the same location.

  REMINDER

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

The vaplocat.ini File

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

By default, the path to this file is:

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

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

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

The maximum length of this name is 20 characters.

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

- VAP SOCKET

  ```
  <LocationName>
  COMM_TYPE=SOCKET
  MONITOR=BVPSCPI
  MESSAGE_LENGTH=31744
  IXO_TIMEOUT=30
  IXO_ADDRESS=127.0.0.1 3000
  ```
- 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
```

- TCP/IP Access TDS/GCOS7

```
vaplocat.ini
<TCP-IP_Access_TDS>
COMM_TYPE=TCPTDSDS>
IXO_HOSTNAME=BC0E
IXO_TDSNAME=VPD5
IXO_PROJECT=PT
IXO_BILLING=PACBASE0
IXO_TERMID=
IXO_TRANSID=VTCP
IXO_DATACONVERT=Y
COMMENT HOST_ENCODING=297
IXO_USERID=PTTD2
```

DETAILS ON THE PARAMETERS

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

- COMM_TYPE:

  This parameter identifies the communication protocol in use.

  The possible values are:

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

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

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

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

- IXO_ADDRESS:

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

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

  Name of RACF group for IMS Connect.
- IXO_TIMEOUT:

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

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

  CICS transaction code
- HOST_ENCODING:

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

  1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MESSAGE_LENGTH:

  This parameter's value must be 31744.
- MONITOR:

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

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

Details of TCP/IP Access TDS/GCOS7 parameters

The following list is the list of required parameters.

- XO_USERID :

  TDS user name.
- IXO_PASSWORD :

GCOS 7 user password.

- IXO_HOSTNAME :

  DPS 7000 host name. The IP address is not allowed.

- IXO_TDSNAME :

  TDS name.

- IXO_TRANSID :

  Communication monitor transaction name.

The following list is the list of optional parameters..

- IXO_PROJECT :

  Project name. Once connected, the client application can start all the transactions allowed for this project, depending on the PROJECT/TDS code indicated in the GCOS7 catalog. If this parameter is composed of spaces only, the default GCOS7 project is taken into account.

- IXO_BILLING :

  The billing is checked in the GCOS 7 catalog. If the parameter is composed of spaces only, the default GCOS 7 billing allocated to the project of this user is taken into account.

- IXO_TERMID :

  Terminal name, not used.

- IXO_DATACONVERT :

  Automatic data conversion, ASCII/EBCDIC. By default, this option is not activated (="N"). If it is activated (="Y"), the ATMI interface automatically manages the ASCII/EBCDIC conversion of the data buffer. The conversion of applicative data can also be managed by the application; example: the middleware Adapters, positioning the HOST_ENCODING=297 parameter. The final message sent by tpsend contains the TDS transaction code, followed by a space and then the applicative data. Therefore, if the automatic conversion is not activated, the IXO layer will manage the conversion of the head part of the message: the transID code and the Space character.

  NB: the IXO_DATACONVERT or HOST_ENCODING conversion options must be exclusive to avoid the double conversion.

- IXO_TIMEOUT (N/A) :

  This delay must be specified in the TIMEOUT key of the Atmi.ini file which is located in the Windows system directory (ex: c:\winnt)

```
[ATMI]
; TIMEOUT : duration for a connection in milliseconds
;             0 : no timeout
TIMEOUT=30000
```

With this TIMEOUT parameter, the tpconnect function can be protected by an expiration delay, to avoid locking situation that can occur in case of incorrect configuration or network or TDS missing response. The default value of the delay is 0, which means that there is no delay mechanism applied.

Uninstalling Client Components

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

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

Chapter 5. Tests

List of Utilities

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

JCL	Description
IVARAD	Archiving of the Administration Database journal
IVSVAD	Saving of the Administration Database
IVROAD	Reorganization of the Administration Database
IVRSAD	Restoration of the Administration Database
IVARCH	Storage of the Development Database journal
IVSAVE	Saving of the Development Database
IVMLIB	Library management
IVREOR	Reorganization of the Development Database
IVREST	Restoration of the Development Database
IVUPDT	Batch update of the Development Database
IVGPRT	Generation print
IVEXLI	Library extraction
IVEXPJ	Journal extraction
IVEXTR	Entities extraction
IVUXSR	sub-networks extraction

Installation Tests

The VA Pac Installation tests include the following operations:

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

TDS submission (TDSRUN)

The TDSRUN procedure supplied, is corresponding to the VA Pac source file alone.

© Copyright IBM Corp. 1983,2003

By default, the '&1' parameter is set to 'STEP1', to load the TPRs in BACKING STORE (VL=STEP1).

For the following times, you must set this parameter to STEP2 to avoid a new TPRs loading.

```
COMM 'VisualAge Pacbase 3.0';
MVL  STEP1,GO='WARM',BASE=$BASE,
     RFTM='DVC=$DVTM,MD=$MDTM';
JUMP &1;
STEP1:
LIB  SM,INLIB1=($TDSNAME.SMLIB);
SYSMAINT COMFILE=*DEMER;
$IN  DEMER;
SM;
LOAD MODULE=TPR  INPUT=INLIB1 REPLACE;
LOAD MODULE=TPR1 INPUT=INLIB1 REPLACE;
LOAD MODULE=TPR2 INPUT=INLIB1 REPLACE;
LOAD MODULE=TPR3 INPUT=INLIB1 REPLACE;
LOAD MODULE=TPR4 INPUT=INLIB1 REPLACE;
LOAD MODULE=TPR5 INPUT=INLIB1 REPLACE;
LOAD MODULE=TPR6 INPUT=INLIB1 REPLACE;
$EIN;
STEP2:
IV   ALPA ($BVP.SYS.PROC),VL=(BASE=&BASE);
IV   ALLB ($BVP.SYS.PROC),VL=(BASE=&BASE);
IV   ALJB ($BVP.SYS.PROC),VL=(BASE=&BASE);
IV   ALSW ($BVP.SYS.PROC),VL=(BASE=&BASE);
IV   ALTR ($BVP.SYS.PROC),VL=(BASE=&BASE);
IV   ALWS ($BVP.SYS.PROC),VL=(BASE=&BASE);
IV   ALHE ($BVP.SYS.PROC),VL=(BASE=&BASE);
JOBLIB SM,$TDSNAME.SMLIB;
STEP $TDSNAME,FILE=($BVP.SYS.PGM),DUMP=DATA,
     DEBUG=($BVP.SYS.PROC,SUBFILE=IL),
     OPTIONS=&GO;
     SZ  250,POOLSZ=70,NBBUF=70;
     ASG H_BJRNL,DVC=$DVTM,MD=$MDTM,TEMPRY,NEXT,POOL;
     DEF H_CTLM,JOURNAL=BEFORE;
     ASG DBUGFILE,$TDSNAME.DEBUG,FILESTAT=CAT,
         SHARE=DIR;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,FILESTAT=CAT,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,FILESTAT=CAT,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE;
     ASG PAC7AY,$BVP.DATA.&BASE.AY,FILESTAT=CAT,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE;
     ASG PAC7AJ,$BVP.DATA.&BASE.AJ,FILESTAT=CAT,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,FILESTAT=CAT,
         ACC=WRITE,SHARE=MONITOR;
     DEF PACGGR,JOURNAL=BEFORE;
```

```
        ASG PACGGN,$BVP.DATA.ADMIN.AN,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PACGGN,JOURNAL=BEFORE;
        ASG PACGGY,$BVP.DATA.ADMIN.AY,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PACGGY,JOURNAL=BEFORE;
        ASG PACGGJ,$BVP.DATA.ADMIN.AJ,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PACGGJ,JOURNAL=BEFORE;
        ASG PAC7AE,$BVP.SYS.AE,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7AE,JOURNAL=BEFORE;
        ASG PAC7DC,$DSMS;
        ASG SYSPAF,$BVP.DATA.&BASE.PA,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF SYSPAF,JOURNAL=BEFORE;
        ASG PACGGU,$BVP.DATA.ADMIN.GU,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PACGGU,JOURNAL=BEFORE;
        ASG PAC7HE,$BVP.DATA.&BASE.HE,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7HE,JOURNAL=BEFORE;
        ASG PAC7LB,$BVP.DATA.&BASE.LB,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7LB,JOURNAL=BEFORE;
        ASG PAC7JB,$BVP.DATA.&BASE.JB,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7JB,JOURNAL=BEFORE;
        ASG PAC7SW,$BVP.DATA.&BASE.SW,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7SW,JOURNAL=BEFORE;
        ASG PAC7TR,$BVP.DATA.&BASE.TR,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7TR,JOURNAL=BEFORE;
        ASG PAC7WS,$BVP.DATA.&BASE.WS,FILESTAT=CAT,
            ACC=WRITE,SHARE=MONITOR;
        DEF PAC7WS,JOURNAL=BEFORE;
    ESTP;
    JUMP CONTINUE;
```

Generation-Print, TP and Batch Update Tests

These tests consist of the following steps:

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

– Executing the 'IVGPRT' JCL (GPRT procedure).

Administration Database Procedures Tests

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

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

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

- Archiving of the journal created during the use tests: execute the 'IVARAD' JCL, which outputs a PJ Admin(1) file.
- Backup of the Administration Database: execute the 'IVSVAD' JCL, which outputs a PC Admin(1) file.
- Reorganization of the sequential backup, PC Admin(1), of the Administration Database: execute the 'IVROAD' JCL, which outputs a PC Admin(2) file.
- Restoration of the Administration Database using the PJ Admin(1) archived transaction file and the PC Admin(2) Database backup file: execute the 'IVRSAD' JCL.

Development Database Procedures Tests

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

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

- Archiving of the journal created during the use tests: execute the 'IVARCH' JCL, which outputs a PJ(1) file.
- Direct backup of the Development Database: execute the 'IVSAVE' JCL, which outputs a PC(1) file.

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

You can then carry on the procedure tests, knowing that the Development Database files must be closed to on-line access. Development Database: execute the 'IVMLIB' JCL, which outputs a PC(2) file.

- Reorganization of the sequential backup, PC(2), of the Development Database: execute the 'IVREOR' JCL, which outputs a PC(3) file.
- Restoration of the Development Database using the PJ(1) archived transaction file and the PC(3) Database backup file: execute the 'IVREST' JCL.

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

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

Extraction-Utility Tests

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

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

- Extraction of a library as transactions: execute the 'IVEXLI' JCL.
- Extraction of entities from a library: execute the 'IVEXTR' JCL.
- Extraction of selected transactions and/or lists of transactions from the archived journal (PJ): execute the 'IVEXPJ' JCL.

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

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

Chapter 6. Re-installation of Server

Standard re-installation - Operations to perform

INTRODUCTION

A reinstallation of the VA Pac system must be done when a new tape of the VA Pac release is delivered, containing improvements or specific developments of the release.

The new version is identified by a number and usually includes:

- A complete installation tape,
- The list of corrected bugs,
- A set of instructions, supplied if the following reinstallation procedure is modified.

Generally, only program libraries, system files and JCLs are affected by the new version.

The reinstallation consists of the execution of most of the procedures used in the first installation.

The complete installation tape is described in Subchapter ″INSTALLATION TAPE″.

The reinstallation includes the following steps:

```
. Installation tape backup,
. Unloading and adapting the JCLs,
. System file reallocation,
. Tape unloading,
. TDS regeneration,
. Program link-edit,
. Taking into account the new VA Pac Error Message file.
```

NOTE:

Do not use reinstallation procedures supplied with earlier versions. Each reinstallation must use the reinstallation procedures supplied, as they are adapted to the contents of the tape being delivered.

UNLOADING AND ADAPTING THE JCLs

© Copyright IBM Corp. 1983,2003

NOTE:: If you have customized JCL, make sure to save them in another library.

The VA Pac JCLs must be re-installed, proceed as follows:

- Unload the JCLs into the library, with the command:

```
LMN SL,IF=(SV.PROC,DVC=CT/M5,MD=$TAPE,FSN=1,END=LEAVE),
    LIB=($BVP.SYS.PROC),
    COM='MV INFILE:^ZZ* REPLACE;';
```

- Replace the former tape label in the ZZVALS member.

  $TAPE=xxxxxx

- Adapt the JCL, by executing ZZJCL.

SYSTEM FILE PRE-ALLOCATION:

During reinstallation, the new version of system files SC, SG, SR, SF, SP, SS, SN and AE0 must be retrieved.

As the size of these files may be modified, they must be re-allocated by 'INPRBS'.

INSTALLATION TAPE UNLOADING:

The unloading, executed by 'INUNLD', is described before.

TDS GENERATION:

A new generation of the TDS is necessary to take into account new versions of the VA Pac routines used in on-line transactions (ZAR100, ZAR200, ZAR400, ZAR980 and ZAR985).

PROGRAM LINK-EDIT:

Execution of the link-edit procedures.

ERROR MESSAGE FILE UPDATE:

The 'INAE' procedure must be executed to take into account the new version of the AE file.

The following JCLs must also be submitted to take into account the possible metamodels improvements:

- IVVING: installation of the administration model (see the description in the appendix, at the end of this document).

- IVVINS: installation of development model extension. This JCL must be submitted for each re-installed database (see the description in the appendix, at the end of this document).

Note

If a reorganization is required in the VING report, the following JCLs should be submitted:

- IVSVAD: job for the administration database save,
- IVROAD: job for the administration database reorganization,
- IVARAD: job for the journal archival of the administration database,
- IVRSAD: job for administration database restoration.

If a reorganization is required in the VINS report, the following JCLs should be submitted:

- IVSAVE: job for the development database save,
- IVREOR: job for the development database reorganization,
- IVARCH: job for the journal archival of the development database,
- IVREST: job for development database restoration.

Chapter 7. Retrieval utilities

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

There are two types of procedures:

Procedures to be executed on the 2.5 database.

They are dedicated to prepare the retrieval in the new release.

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

These procedures are :

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

© Copyright IBM Corp. 1983,2003

Chapter 8. Retrieval

Retrieval of VisualAge Pacbase 2.0 and 2.5

Operations to be Performed

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

SCREEN BRANCHING:

It consists of six steps:

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

- SAVE: backup of the Development Database (PC),
- PARM: backup of the user parameters (PE),
- SVAG: backup of the generation-print commands (PG)
- SVPE: backup of the PEI environment (PP).
- 2) Installation of the 3.0 Administration Database

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

  It will create the GN, GR, GY, GJ and GU files.
- You must execute the following JCLs:
  - creation of the Administration Database,
    - INGU: creation and initialization of GU user codes file,
    - RSAD: initialization of the Administration Database with installation data (enter the access key),
    - INQJ: initialization of the QJ archive file
    - VING: installation of the Administration Model (see the Appendix, at the end of the manual),
  - re-organization of the Administration Database if it is mentioned in the execution report of the preceding job,
    - SVAD: backup of the Administration Database,
    - ROAD: re-organization of the Administration Database,
    - ARAD: initialization of the Administration Database journal file,
    - RSAD: restoration of the Administration Database,

© Copyright IBM Corp. 1983,2003

– retrieval of the old Database data,
  - PE25: retrieval of user parameters from the PE file which was generated during step 1,
  - SVAD: backup of the Administration Database.

3) Retrieval of a Development Database.

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

It consists in executing the following JCLs:
– PC25: retrieval 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 JCL's is optional and it is sometimes required for a better optimization.
– SAVE: 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 involves executing the following JCL:
– PG20: if retrieval of 2.0 AG file,
– PG25: if retrieval of 2.5 AG file.

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

It involves in executing the following JCL's:
– UV25: retrieval of the UV file data.

6) Retrieval of PEI files.

It involves executing the following JCL:
– 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

– the transaction code: it is used to connect to the Database in character mode. The $BASE installation parameter is also used to code the file names.

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

The 'B' line has the following structure:

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

PE25 - Description of Steps

PROCESSING OF USER PARAMETERS (PE): PTU920

Code	Physical name	Type	Label
PAC7EN	&OLDPE	Input	User parameters, old version
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7MB	TMBPE25	Input	User input
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGY	Output	User parameter transactions (length=310)
PAC7ET		Report	Error report

TRANSACTION FORMATTING: PAF900

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PAC7AN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PAC7AE	$BVP.SYS.AE	Input	Error labels
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index

Code	Physical name	Type	Label
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGY	Input	Update transactions
PAC7MV	TMV	Output	Transactions formatting (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	TME	Output	Work file (length=372)
PAC7MW	TMW	Output	Work file (length=170)
PAC7MX	TMX	Output	Work file (length=743)
PAC7MY	TMY	Output	Work file (length=743)

UPDATE OF THE ADMINISTRATION DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Output	Administration Database Data file
PAC7AN	$BVP.DATA.ADMIN.AN	Output	Administration Database Index file
PAC7AY	$BVP.DATA.ADMIN.AY	Output	Administration Database extension
PAC7AJ	$BVP.DATA.ADMIN.AJ	Output	Administration Database journal
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index file
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data file
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	TME	Input	Work file
PAC7MV	TMV	Input	Update transactions
PAC7RB	TRB	Output	UPDT erroneous transactions (length=80)

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

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

Return codes:

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

PE25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                 RETRIEVAL OF PE FILE                  ';
COMM '                                                       ';
COMM ' ----------------------------------------------------- ';
COMM '                                                       ';
MVL  SZMB=5,SZWK=5,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PE25,
     OF=(TMBPE25,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP BVPTU920,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7EN,&OLDPE;
     ASG PAC7MB,TMBPE25,TEMPRY,&RFTM,END=PASS;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM,END=PASS;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
STEP BVPAF900,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
```

```
          ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
          ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
          ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
          ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
          ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
          ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM;
     ALC PAC7ME,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7ME,TME,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MW,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TMW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MX,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MX,TMX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MY,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MY,TMY,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPACA15,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
          ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.&BASE.AJ,
          ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
          ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
     ASG PAC7DC,$DSMS;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
          ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7AY,$BVP.DATA.&BASE.AY,
          ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7MV,TMV,TEMPRY,&RFTM;
     ASG PAC7ME,TME,TEMPRY,&RFTM;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG H_BJRNL,FILESTAT=TEMPRY,
          DVC=$DVTM,MD=$MDTM;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=WRITE,SHARE=MONITOR;
```

```
     DEF PACGGN,JOURNAL=BEFORE;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGR,JOURNAL=BEFORE;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGY,JOURNAL=BEFORE;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGU,JOURNAL=BEFORE;
     ASG PAC7RB,TRB,TEMPRY,&RFTM;
     ASG PAC7RY,TRY,TEMPRY,&RFTM;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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	&OLDPC	Input	Sequential image of the network (old release)
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7MB	TMBPC25	Input	User input
PAC7PB	TPC	Output	First data record (length=153)

Code	Physical name	Type	Label
PAC7PE	TPE	Output	User Entity Occurrence definition (2.5 release), Report layouts, and Comments (except the calls of Input Aids) (length=193)
PAC7PG	TPG	Output	Description of Input Aids and Comments including calls of Input Aids. (length=193)
PAC7PL	TPL	Output	Definition and Description of Volumes, Definition and Description of Manuals (length=193)
PAC7PZ	TPZ	Output	User Entities and description of their Occurrences (2.5 release)
PAC7PF	TPF	Output	Other records (length=153)
PAC7PM	TPM	Output	Report file (length=48)
PAC7ET		Report	Report only if absence of Database code

MANUALS AND VOLUMES PROCESSING: PTU909

Code	Physical name	Type	Label
SWK		Sort	
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7PB	TPC	Input	First data record
PAC7PL	TPL	Input	Definition and Description of Volumes and Manuals
PAC7PI	TPI	Output	Sorted and re-formatted Volumes Definitions and Descriptions (length=153)
PAC7PM	TPM	Input/Output	Report file

COMMENTS PROCESSING: PTU92A

Code	Physical name	Type	Label
SWK		Sort	
PAC7AE	$BVP.SYS.AE	Input	Error messages

Code	Physical name	Type	Label
PAC7PG	TPG	Input	Description of Input Aids and of the call of Input Aids in the Comments
PAC7PM	TPM	Input/Output	Report
PAC7PE	TPH	Output	Description of Input Aids and of the call of Input Aids in Comments (length=193)

META-ENTITIES PROCESSING: PTU912

Code	Physical name	Type	Label
SWK		Sort	
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7PZ	TPZ	Input	User Entities (2.5 release)
PAC7PB	TPB	Input	First data record
PAC7ZP	TZP	Output	Development Model records (Definition and Descriptions) (length=193)

REPORT LAYOUT PROCESSING: PTU919

Code	Physical name	Type	Label
SWK		Sort	
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7PE	TPE	Input	User Entity Occurrences Definition (2.5 rel.), Report layouts and Comments (except calls of input aids)
PAC7PB	TPC	Input	First data record
PAC7PH	TPH	Input	Description of Input Aids and of their calls in the Comments
PAC7PM	TPM	Input/Output	Report file
PAC7ZP	TEP	Output	User entity Occurrences Definition (2.5 rel.), Report layouts, and Comments (call of Input Aids included) (length=193)
PAC7PO	TPO	Output	Comments (except the call of Input Aids) (length=153)

Code	Physical name	Type	Label
PAC7PD	TPD	Output	First data record (length=153)

USER ENTITIES PROCESSING: PTU913

Code	Physical name	Type	Label
SWK		Sort	
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7PX	TEP	Input	Definition of User Entity Occurrences (2.5 release), Report layouts, and Comments (including the call of Input Aids)
PAC7PZ	TZP	Input	Definition and Description of the Development Model and Description of User Entity Occurrences (2.5 release)
PAC7PB	TPD	Input	First data record
PAC7ZP	TZX	Output	Long data of the Development Model, User Entities, Report layouts, and Comments (including the calls of Input Aids) (length=193)
PAC7PD	TPB	Output	First data record (length=153)

SORT OF LONG DATA

Sort criteria: SRTPC25 member of the SY PDS

Code	Physical name	Type	Label
SWK		Sort	
SORTIN	TZX	Input	Intermediate long data
SORTOUT	TZX	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	$BVP.SYS.AE	Input	Error messages

Code	Physical name	Type	Label
PAC7ZP	TZX	Input	Sorted long data
PAC7PO	TPO	Input	Comments (no call of Input Aids)
PAC7PD	TPB	Input	First data record
PAC7PI	TPI	Input	Volumes Definition and Description
PAC7PF	TPF	Input	Other records
PAC7PM	TPM	Input	Report file
PAC7PC	$BVP.DATA.&BASE.PC/G+1	Output	Sequential image of the network (3.0 release)
PAC7ET		Report	Retrieval report

PC25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                 RETRIEVAL OF PC FILE                  ';
COMM '                                                       ';
COMM ' ----------------------------------------------------- ';
COMM '                                                       ';
MVL  OLDPC='OLDPC',PAC7PC='$BVP.DATA.'&BASE'.PC',
     SZET=5,SZWK=5,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PC25,
     OF=(TMBPC25,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP BVPTU911,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7MC,&OLDPC;
     ASG PAC7MB,TMBPC25,TEMPRY,&RFTM,END=PASS;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PB,TPC,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PB,SZ=1,UNIT=TRK,INCRSZ=1;
     ASG PAC7PE,TPE,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PE,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PE,CISIZE=$CISEQ,NBBUF=1;
     ASG PAC7PF,TPF,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PF,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PF,CISIZE=$CISEQ,NBBUF=1;
     ASG PAC7PG,TPG,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PG,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PG,CISIZE=$CISEQ,NBBUF=1;
     ASG PAC7PL,TPL,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PL,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PL,CISIZE=$CISEQ,NBBUF=1;
     ASG PAC7PM,TPM,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PM,SZ=&SZET,UNIT=CYL,INCRSZ=2;
```

```
     DEF PAC7PM,CISIZE=$CISEQ,NBBUF=1;
     ASG PAC7PZ,TPZ,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PZ,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PZ,CISIZE=$CISEQ,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPTU909,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PB,TPC,TEMPRY,&RFTM,END=PASS;
     ASG PAC7PL,TPL,TEMPRY,&RFTM;
     ASG PAC7PM,TPM,TEMPRY,&RFTM,END=PASS;
     ASG PAC7PI,TPI,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PI,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PI,CISIZE=$CISEQ,NBBUF=1;
     SWK WKDISK=(SZ=2,&RFTM);
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPTU92A,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PG,TPG,TEMPRY,&RFTM;
     ASG PAC7PM,TPM,TEMPRY,&RFTM,END=PASS;
     ASG PAC7PE,TPH,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PE,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PE,CISIZE=14336,NBBUF=1;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
STEP BVPTU912,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PZ,TPZ,TEMPRY,&RFTM;
     ASG PAC7PB,TPC,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ZP,TZP,TEMPRY,&RFTM,END=PASS;
     ALC PAC7ZP,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7ZP,CISIZE=14336,NBBUF=1;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
STEP BVPTU919,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PE,TPE,TEMPRY,&RFTM;
     ASG PAC7PB,TPC,TEMPRY,&RFTM;
     ASG PAC7PH,TPH,TEMPRY,&RFTM;
     ASG PAC7PM,TPM,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ZP,TEP,TEMPRY,&RFTM,END=PASS;
     ALC PAC7ZP,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7ZP,CISIZE=14336,NBBUF=1;
```

```
     ASG PAC7PO,TPO,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PO,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PO,CISIZE=14336,NBBUF=1;
     ASG PAC7PD,TPD,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PD,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PD,CISIZE=14336,NBBUF=1;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
STEP BVPTU913,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PZ,TZP,TEMPRY,&RFTM;
     ASG PAC7PB,TPD,TEMPRY,&RFTM;
     ASG PAC7PX,TEP,TEMPRY,&RFTM;
     ASG PAC7PD,TPB,TEMPRY,&RFTM,END=PASS;
     ALC PAC7PD,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7PD,CISIZE=14336,NBBUF=1;
     ASG PAC7ZP,TZX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7ZP,SZ=&SZET,UNIT=CYL,INCRSZ=2;
     DEF PAC7ZP,CISIZE=14336,NBBUF=1;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
SORT IF=(TZX,TEMPRY,&RFTM,END=PASS),
     OF=(INFILE),
     WKDISK=(SZ=&SZWK,&RFTM),
     COMFILE=($BVP.SYS.PROC,,SUBFILE=SRTPC25);
STEP BVPTU914,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7PF,TPF,TEMPRY,&RFTM;
     ASG PAC7PD,TPB,TEMPRY,&RFTM;
     ASG PAC7PI,TPI,TEMPRY,&RFTM;
     ASG PAC7PM,TPM,TEMPRY,&RFTM;
     ASG PAC7PO,TPO,TEMPRY,&RFTM;
     ASG PAC7PC,&PAC7PC!!/G+1;
     ASG PAC7ZP,TZX,TEMPRY,&RFTM;
     ASG PAC7ET,SYS.OUT;
ESTP;
SHIFT &PAC7PC;
JUMP END;
```

Generation-Print Commands Retrieval (PG20)

PG20 - Introduction

PRINCIPLE

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

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

EXECUTION CONDITIONS

The files of the administration and development Databases must be closed in the on-line mode.

PRINTED OUTPUT

This procedure outputs a report which contains the errors encountered.

NOTE

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

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

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

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

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

PG20 - Input / Processing / Results

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

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

The line structure is as follows:

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

PG20 - Description of Steps

GENERATION-PRINT COMMANDS FORMATTING: PTU908

Code	Physical name	Type	Label
PAC7IN	&OLDPG	Input	Generation-print commands, old release
PAC7OU	TPG	Output	Re-formatted generation-print commands (length=150)

GENERATION-PRINT COMMANDS PROCESSING: PTU921

Code	Physical name	Type	Label
PAC7PG	TPG	Input	Generation-print commands, old release
PAC7AE	$BVP.SYS.AE	Input	Error labels
PAC7MB	TMBPG20	Input	User Entities
PAC7GY	TGY	Output	Generation-print commands transactions (length=310)
PAC7GZ	TGZ	Output	JCL lines transactions (length = 310)
PAC7ET		Report	Error report

TRANSACTIONS FORMATTING: PAF900

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development Database data
PAC7AN	$BVP.DATA.&BASE.AN	Input	Development Database index
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGY	Input	Update transactions

Code	Physical name	Type	Label
PAC7MV	TMV	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	TME	Output	Work file (length=372)
PAC7MW	TMW	Output	Work file (length=170)
PAC7MX	TMX	Output	Work file (length=743)
PAC7MY	TMY	Output	Work file (length=743)

UPDATE OF THE DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.&BASE.AR	Output	Development Database Data file
PAC7AN	$BVP.DATA.&BASE.AN	Output	Development Database index
PAC7AY	$BVP.DATA.&BASE.AY	Output	Development Database extension
PAC7AJ	$BVP.DATA.&BASE.AJ	Output	Development Database journal
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index file
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data file
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database Elements
PAC7ME	TME	Input	Work file
PAC7MV	TMV	Input	Update transactions
PAC7RB	TRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	TRY	Output	UPDP erroneous transactions (length=310)

Code	Physical name	Type	Label
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

Transactions formatting : PAF900

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PAC7AN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGZ	Input	Update transactions
PAC7MV	TMV	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	TME	Output	Work file (length=372)
PAC7MW	TMW	Output	Work file (length=170)
PAC7MX	TMX	Output	Work file (length=743)
PAC7MY	TMY	Output	Work file (length=743)

UPDATE OF THE ADMINISTRATION DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Output	Administration Database Data file
PAC7AN	$BVP.DATA.ADMIN.AN	Output	Administration Database Index file
PAC7AY	$BVP.DATA.ADMIN.AY	Output	Administration Database extension
PAC7AJ	$BVP.DATA.ADMIN.AJ	Output	Administration Database journal
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index file
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data file
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	TME	Input	Work file
PAC7MV	TMV	Input	Update transactions
PAC7RB	TRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	TRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

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

Return codes:

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

PG20 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '             RETRIEVAL OF PG FILE SINCE 2.0             ';
COMM '                                                        ';
COMM ' ------------------------------------------------------ ';
COMM '                                                        ';
MVL  OLDPG='OLDPG',
     SZT=1,SZWK=5,SZMB=5,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PG20,
     OF=(TMBPG20,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP BVPTU908,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7OU,TPG,TEMPRY,&RFTM,END=PASS;
     ALC PAC7OU,SZ=&SZT,UNIT=CYL,INCRSZ=2;
     ASG PAC7IN,&OLDPG;
ESTP;
STEP BVPTU921,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7MB,TMBPG20,TEMPRY,&RFTM,END=PASS;
     ASG PAC7PG,TPG,TEMPRY,&RFTM;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GY,SZ=&SZT,UNIT=CYL,INCRSZ=2;
     ASG PAC7GZ,TGZ,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GZ,SZ=&SZT,UNIT=CYL,INCRSZ=2;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
STEP BVPAF900,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM;
     ALC PAC7ME,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7ME,TME,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
```

```
     ASG PAC7MB,TMBPG20,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MW,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TMW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MX,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MX,TMX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MY,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MY,TMY,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPACA15,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.&BASE.AJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
     ASG PAC7DC,$DSMS;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7AY,$BVP.DATA.&BASE.AY,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7MV,TMV,TEMPRY,&RFTM;
     ASG PAC7ME,TME,TEMPRY,&RFTM;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG H_BJRNL,FILESTAT=TEMPRY,
         DVC=$DVTM,MD=$MDTM;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGY,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7RB,TRB,TEMPRY,&RFTM;
     ASG PAC7RY,TRY,TEMPRY,&RFTM;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPAF900,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
```

```
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7GY,TGZ,TEMPRY,&RFTM;
     ALC PAC7ME,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7ME,TME,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MB,TMBPG20,TEMPRY,&RFTM;
     ALC PAC7MW,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TMW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MX,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MX,TMX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MY,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MY,TMY,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPACA15,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.ADMIN.AJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.ADMIN.AN,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
     ASG PAC7DC,$DSMS;
     ASG PAC7AR,$BVP.DATA.ADMIN.AR,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7AY,$BVP.DATA.ADMIN.AY,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7MV,TMV,TEMPRY,&RFTM;
     ASG PAC7ME,TME,TEMPRY,&RFTM;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG H_BJRNL,FILESTAT=TEMPRY,
         DVC=$DVTM,MD=$MDTM;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGN,JOURNAL=BEFORE;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGR,JOURNAL=BEFORE;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGY,JOURNAL=BEFORE;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGU,JOURNAL=BEFORE;
```

```
     ASG PAC7RB,TRB,TEMPRY,&RFTM;
     ASG PAC7RY,TRY,TEMPRY,&RFTM;
ESTP;
JUMP ERR,SW20,EQ,1;
```

Generation-Print Commands Retrieval (PG25)

PG25 - Introduction

PRINCIPLE

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

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

EXECUTION CONDITIONS

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

PRINTED OUTPUT

This procedure prints a report on the errors encountered.

NOTE

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

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

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

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

PG25 - Input / Processing / Results

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

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

The line structure is as follows:

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

PG25 - Description of Steps

GENERATION-PRINT COMMANDS PROCESSING: PTU921

Code	Physical name	Type	Label
PAC7PG	1OLDPG	Input	Generation-print commands, old release
PAC7AE	$BVP.SYS.AE	Input	Error labels
PAC7MB	TMBPG25	Input	User Entities
PAC7GY	TGY	Output	Generation-print commands transactions (length=310)
PAC7GZ	TGZ	Output	JCL lines transactions (length = 310)
PAC7ET		Report	Error report

TRANSACTIONS FORMATTING: PAF900

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development Database data
PAC7AN	$BVP.DATA.&BASE.AN	Input	Development Database index
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGY	Input	Update transactions

Code	Physical name	Type	Label
PAC7MV	TMV	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	TME	Output	Work file (length=372)
PAC7MW	TMW	Output	Work file (length=170)
PAC7MX	TMX	Output	Work file (length=743)
PAC7MY	TMY	Output	Work file (length=743)

UPDATE OF THE DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.&BASE.AR	Output	Development Database Data file
PAC7AN	$BVP.DATA.&BASE.AN	Output	Development Database index
PAC7AY	$BVP.DATA.&BASE.AY	Output	Development Database extension
PAC7AJ	$BVP.DATA.&BASE.AJ	Output	Development Database journal
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index file
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data file
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database Elements
PAC7ME	TME	Input	Work file
PAC7MV	TMV	Input	Update transactions
PAC7RB	TRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	TRY	Output	UPDP erroneous transactions (length=310)

Code	Physical name	Type	Label
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

Transactions formatting : PAF900

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PAC7AN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGZ	Input	Update transactions
PAC7MV	TMV	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	TME	Output	Work file (length=372)
PAC7MW	TMW	Output	Work file (length=170)
PAC7MX	TMX	Output	Work file (length=743)
PAC7MY	TMY	Output	Work file (length=743)

UPDATE OF THE ADMINISTRATION DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Output	Administration Database Data file
PAC7AN	$BVP.DATA.ADMIN.AN	Output	Administration Database Index file
PAC7AY	$BVP.DATA.ADMIN.AY	Output	Administration Database extension
PAC7AJ	$BVP.DATA.ADMIN.AJ	Output	Administration Database journal
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index file
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data file
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	TME	Input	Work file
PAC7MV	TMV	Input	Update transactions
PAC7RB	TRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	TRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

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

Return codes:

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

PG25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '             RETRIEVAL OF PG FILE SINCE 2.5             ';
COMM '                                                        ';
COMM ' ------------------------------------------------------ ';
COMM '                                                        ';
MVL  OLDPG='OLDPG',
     SZT=1,SZWK=5,SZMB=5,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PG25,
     OF=(TMBPG25,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP BVPTU921,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7MB,TMBPG25,TEMPRY,&RFTM,END=PASS;
     ASG PAC7PG,&OLDPG;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GY,SZ=&SZT,UNIT=CYL,INCRSZ=2;
     ASG PAC7GZ,TGZ,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GZ,SZ=&SZT,UNIT=CYL,INCRSZ=2;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END,SW30,EQ,1;
STEP BVPAF900,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM;
     ALC PAC7ME,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7ME,TME,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MW,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TMW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MX,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MX,TMX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MY,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
```

```
     ASG PAC7MY,TMY,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPACA15,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.&BASE.AJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
     ASG PAC7DC,$DSMS;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7AY,$BVP.DATA.&BASE.AY,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7MV,TMV,TEMPRY,&RFTM;
     ASG PAC7ME,TME,TEMPRY,&RFTM;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG H_BJRNL,FILESTAT=TEMPRY,
         DVC=$DVTM,MD=$MDTM;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGY,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7RB,TRB,TEMPRY,&RFTM;
     ASG PAC7RY,TRY,TEMPRY,&RFTM;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPAF900,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
```

```
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7GY,TGZ,TEMPRY,&RFTM;
     ALC PAC7ME,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7ME,TME,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MW,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TMW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MX,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MX,TMX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MY,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MY,TMY,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPACA15,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.ADMIN.AJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.ADMIN.AN,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
     ASG PAC7DC,$DSMS;
     ASG PAC7AR,$BVP.DATA.ADMIN.AR,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7AY,$BVP.DATA.ADMIN.AY,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7MV,TMV,TEMPRY,&RFTM;
     ASG PAC7ME,TME,TEMPRY,&RFTM;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG H_BJRNL,FILESTAT=TEMPRY,
         DVC=$DVTM,MD=$MDTM;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGN,JOURNAL=BEFORE;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGR,JOURNAL=BEFORE;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGY,JOURNAL=BEFORE;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=WRITE,SHARE=MONITOR;
     DEF PACGGU,JOURNAL=BEFORE;
     ASG PAC7RB,TRB,TEMPRY,&RFTM;
     ASG PAC7RY,TRY,TEMPRY,&RFTM;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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	&OLDPP	Input	Back up of production environment (old release)
PAC7AE	$BVP.SYS.AE	Input	Error message
PAC7MB	TMBPP25	Input	User input
PAC7GY	TGY	Output	Records of production environments (length=310)
PAC7ET		Report	Report in case of error

TRANSACTIONS FORMATTING: PAF900

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development Database data
PAC7AN	$BVP.DATA.&BASE.AN	Input	Development Database index
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGY	Input	Update transactions
PAC7MV	TMV	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	TME	Output	Work file (length=372)
PAC7MW	TMW	Output	Work file (length=170)
PAC7MX	TMX	Output	Work file (length=743)
PAC7MY	TMY	Output	Work file (length=743)

UPDATE OF THE DEVELOPMENT DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.&BASE.AR	Output	Development Database Data file
PAC7AN	$BVP.DATA.&BASE.AN	Output	Development Database index
PAC7AY	$BVP.DATA.&BASE.AY	Output	Development Database extension
PAC7AJ	$BVP.DATA.&BASE.AJ	Output	Development Database journal
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index file
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data file
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database Elements
PAC7ME	TME	Input	Work file
PAC7MV	TMV	Input	Update transactions
PAC7RB	TRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	TRY	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

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

Return codes:

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

PP25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '             RETRIEVAL OF PP FILE                        ';
COMM '                                                         ';
```

```
COMM ' ------------------------------------------------------ ';
COMM '                                                        ';
MVL  OLDPP='OLDPP',
     SZMB=5,SZWK=5,SZT=1,
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*PP25,
     OF=(TMBPP25,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP BVPTU923,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7MB,TMBPP25,TEMPRY,&RFTM,END=PASS;
     ASG PAC7PP,&OLDPP;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GY,SZ=&SZT,UNIT=CYL,INCRSZ=2;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPAF900,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM;
     ALC PAC7ME,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7ME,TME,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MB,TMBPP25,TEMPRY,&RFTM;
     ALC PAC7MW,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TMW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MX,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MX,TMX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MY,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MY,TMY,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPACA15,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
```

```
        ACC=READ,SHARE=MONITOR;
    DEF PAC7AE,READLOCK=STAT;
    ASG PAC7AJ,$BVP.DATA.&BASE.AJ,
        ACC=WRITE,SHARE=MONITOR;
    DEF PAC7AJ,JOURNAL=BEFORE;
    ASG PAC7AN,$BVP.DATA.&BASE.AN,
        ACC=WRITE,SHARE=MONITOR;
    DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
    ASG PAC7DC,$DSMS;
    ASG PAC7AR,$BVP.DATA.&BASE.AR,
        ACC=WRITE,SHARE=MONITOR;
    DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
    ASG PAC7AY,$BVP.DATA.&BASE.AY,
        ACC=WRITE,SHARE=MONITOR;
    DEF PAC7AY,JOURNAL=BEFORE,NBBUF=4;
    ASG PAC7MV,TMV,TEMPRY,&RFTM;
    ASG PAC7ME,TME,TEMPRY,&RFTM;
    ASG PAC7IE,SYS.OUT;
    ASG PAC7IF,SYS.OUT;
    ASG H_BJRNL,FILESTAT=TEMPRY,
        DVC=$DVTM,MD=$MDTM;
    ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=READ,SHARE=MONITOR;
    DEF PACGGN,READLOCK=STAT;
    ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=READ,SHARE=MONITOR;
    DEF PACGGR,READLOCK=STAT;
    ASG PACGGY,$BVP.DATA.ADMIN.AY,ACC=READ,SHARE=MONITOR;
    DEF PACGGY,READLOCK=STAT;
    ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=READ,SHARE=MONITOR;
    DEF PACGGU,READLOCK=STAT;
    ASG PAC7RB,TRB,TEMPRY,&RFTM;
    ASG PAC7RY,TRY,TEMPRY,&RFTM;
ESTP;
JUMP ERR,SW20,EQ,1;
```

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	&OLDUV	Input	Transfer parameters, old release
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7MB	TMBUV25	Input	User input
PAC7GY	TGY	Output	Transfer parameter transactions (length=310)
PAC7ET		Report	Report in case of error

TRANSACTION FORMATTING: PAF900

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PAC7AN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PAC7AE	$BVP.SYS.AE	Input	Error labels
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database data
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7GY	TGY	Input	Update transactions

Code	Physical name	Type	Label
PAC7MV	TMV	Output	Transactions formatting (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME	TME	Output	Work file (length=372)
PAC7MW	TMW	Output	Work file (length=170)
PAC7MX	TMX	Output	Work file (length=743)
PAC7MY	TMY	Output	Work file (length=743)

UPDATE OF THE ADMINISTRATION DATABASE: PACA15

Code	Physical name	Type	Label
PAC7AR	$BVP.DATA.ADMIN.AR	Output	Administration Database Data file
PAC7AN	$BVP.DATA.ADMIN.AN	Output	Administration Database Index file
PAC7AY	$BVP.DATA.ADMIN.AY	Output	Administration Database extension
PAC7AJ	$BVP.DATA.ADMIN.AJ	Output	Administration Database journal
PAC7AE	$BVP.SYS.AE	Input	Error messages
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index file
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data file
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database users
PAC7DC	DUMMY	Input	DSMS file of Development Database elements
PAC7ME	TME	Input	Work file
PAC7MV	TMV	Input	Update transactions
PAC7RB	TRB	Output	UPDT erroneous transactions (length=80)
PAC7RY	TRY	Output	UPDP erroneous transactions (length=310)

Code	Physical name	Type	Label
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	Summary of erroneous transactions (length=132)

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

Return codes:

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

UV25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                 RETRIEVAL OF UV FILE                  ';
COMM '                                                       ';
COMM ' ----------------------------------------------------- ';
COMM '                                                       ';
MVL  OLDUV='OLDUV',
     RFTM='DVC=$DVTM,MD=$MDTM';
CR   IF=*UV25,
     OF=(TMBUV25,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISZ=2048,RECSZ=80,RECFORM=FB);
STEP BVPTU922,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7UV,&OLDUV;
     ASG PAC7MB,TMBUV25,TEMPRY,&RFTM,END=PASS;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM,END=PASS;
     ALC PAC7GY,SZ=100,UNIT=TRK,INCRSZ=1;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPAF900,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,NBBUF=4;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
```

```
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7GY,TGY,TEMPRY,&RFTM;
     ALC PAC7ME,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7ME,TME,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MV,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     ASG PAC7MB,TMBUV25,TEMPRY,&RFTM;
     ALC PAC7MW,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MW,TMW,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MX,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MX,TMX,TEMPRY,&RFTM,END=PASS;
     ALC PAC7MY,SZ=&SZMB,UNIT=CYL,INCRSZ=1;
     ASG PAC7MY,TMY,TEMPRY,&RFTM,END=PASS;
     ASG PAC7EI,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
STEP BVPACA15,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.&BASE.AJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE,NBBUF=10;
     ASG PAC7DC,$DSMS;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7AY,$BVP.DATA.&BASE.AY,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE,NBBUF=4;
     ASG PAC7MV,TMV,TEMPRY,&RFTM;
     ASG PAC7ME,TME,TEMPRY,&RFTM;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG H_BJRNL,FILESTAT=TEMPRY,
         DVC=$DVTM,MD=$MDTM;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,ACC=READ,SHARE=MONITOR;
     DEF PACGGY,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7RB,TRB,TEMPRY,&RFTM;
     ASG PAC7RY,TRY,TEMPRY,&RFTM;
ESTP;
```

```
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' UV25 - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

Retrieval of MB Transactions (MB25)

MB25 - Introduction

PRINCIPLE

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

REQUISITES

None.

PRINTED OUTPUT

This procedure prints a report of the errors encountered.

RESULT

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

MB25 - Description of Steps

MB FILE RETRIEVAL: PTU926

Code	Physical name	Type	Label
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development database data
PACGGR	$BVP.DATA.ADMIN.GR	Input	Administration database data
PACGGN	$BVP.DATA.ADMIN.GN	Input	Administration database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration database users
PAC7MB	&OLDMB	Input	2.5 MB transactions
PAC7MV	TMV	Output	Retrieval transactions for UPDT
PAC7ME	TMR	Output	Revoked transactions
PAC7EF		Report	Retrieval reports

Code	Physical name	Type	Label
PAC7DD		Report	Batch procedure authorization option

MB25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                RETRIEVAL OF MB FILE                    ';
COMM '                                                        ';
COMM ' ------------------------------------------------------ ';
COMM '                                                        ';
MVL  OLDMB='OLDMB',
     RFTM='DVC=$DVTM,MD=$MDTM';
STEP BVPTU926,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7MB,&OLDMB;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7ME,TMR,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ME,NBBUF=1;
     ASG PAC7EF,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
LMN  SL INFILE=(TMV,TEMPRY,&RFTM),
     LIB=($BVP.DATA.&BASE.USERS),
     COM='MV INFILE:MBUPDT_MB25'&USER',INFORM=SARF,
     TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
JUMP END;
ERR:
SEND '  - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

Retrieval of GY Transactions (GY25)

GY25 - Introduction

PRINCIPLE

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

REQUISITES

None.

PRINTED OUTPUT

This procedure prints a report of the errors encountered.

RESULT

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

GY25 - Description of Steps

GY FILE RETRIEVAL: PTU927

Code	Physical name	Type	Label
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development database data
PACGGR	$BVP.DATA.ADMIN.GR	Input	Administration database data
PACGGN	$BVP.DATA.ADMIN.GN	Input	Administration database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration database users
PAC7GY	&OLDGY	Input	2.5 GY transactions
PAC7MV	TMV	Output	Retrieval transactions for UPDP
PAC7ME	TMR	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

GY25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                  RETRIEVAL OF GY FILE                  ';
COMM '                                                        ';
COMM ' ------------------------------------------------------ ';
COMM '                                                        ';
MVL  OLDMB='OLDGY',
     RFTM='DVC=$DVTM,MD=$MDTM';
STEP BVPTU927,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7GY,&OLDGY;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
```

```
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7ME,TMR,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ME,NBBUF=1;
     ASG PAC7EF,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
LMN  SL INFILE=(TMV,TEMPRY,&RFTM),
     LIB=($BVP.DATA.&BASE.USERS),
     COM='MV INFILE:MBUPDT_GY25'&USER',INFORM=SARF,
     TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
JUMP END;
ERR:
SEND '  - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

Retrieval of MB Transactions (MB30)

MB30 - Introduction

PRINCIPLE

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

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	$BVP.SYS.AE	Input	Error messages
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development database data
PACGGR	$BVP.DATA.ADMIN.GR	Input	Administration database data
PACGGN	$BVP.DATA.ADMIN.GN	Input	Administration database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration database users
PAC7MB	&OLDMB	Input	3.0 MB transactions
PAC7MV	TMV	Output	Retrieval transactions for UPDT
PAC7ME	TMR	Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

MB30 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                 RETRIEVAL OF MB FILE                  ';
COMM '                                                       ';
COMM ' -------------------------------------------------------';
COMM '                                                       ';
MVL  OLDMB='OLDMB',
     RFTM='DVC=$DVTM,MD=$MDTM';
STEP BVPTU928,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7MB,&OLDMB;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7ME,TMR,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ME,NBBUF=1;
     ASG PAC7EF,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
LMN  SL INFILE=(TMV,TEMPRY,&RFTM),
     LIB=($BVP.DATA.&BASE.USERS),
```

```
     COM='MV INFILE:MBUPDT_MB30'&USER',INFORM=SARF,
     TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
JUMP END;
ERR:
SEND '  - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

Retrieval of GY Transactions (GY30)

GY30 - Introduction

PRINCIPLE

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

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	$BVP.SYS.AE	Input	Error messages
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development database data
PACGGR	$BVP.DATA.ADMIN.GR	Input	Administration database data
PACGGN	$BVP.DATA.ADMIN.GN	Input	Administration database index
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration database users
PAC7GY	&OLDGY	Input	3.0 GY transactions
PAC7MV	TMV	Output	Retrieval transactions for UPDP
PAC7ME	TMR	Output	Revoked transactions
PAC7EF		Report	Retrieval reports

Code	Physical name	Type	Label
PAC7DD		Report	Authorization option

GY30 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                 RETRIEVAL OF GY FILE                  ';
COMM '                                                       ';
COMM ' -------------------------------------------------------';
COMM '                                                       ';
MVL  OLDMB='OLDGY',
     RFTM='DVC=$DVTM,MD=$MDTM';
STEP BVPTU929,FILE=$BVP.SYS.PGM,REPEAT,DUMP=DATA;
     ASG PAC7GY,&OLDGY;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,ACC=READ,SHARE=MONITOR;
     DEF PAC7AR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PAC7MV,TMV,TEMPRY,&RFTM,END=PASS;
     DEF PAC7MV,NBBUF=1;
     ASG PAC7ME,TMR,TEMPRY,&RFTM,END=PASS;
     DEF PAC7ME,NBBUF=1;
     ASG PAC7EF,SYS.OUT;
     ASG PAC7DD,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
LMN  SL INFILE=(TMV,TEMPRY,&RFTM),
     LIB=($BVP.DATA.&BASE.USERS),
     COM='MV INFILE:MBUPDT_GY30'&USER',INFORM=SARF,
     TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP ERR,SEV,GE,3;
JUMP END;
ERR:
SEND '  - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

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, the number of transactions converted in a 3.0 format, and the number of written transactions.

WARNING: The number of written transactions can be higher than the number of retrieved transactions. From an old 2.n transaction, several 3.0 transactions can be created, in particular for meta entities and user entities.

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	&OLDPJ	Input	Journal file old version
PAC7AE	$BVP.SYS.AE	Input	Error messages
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development Database data
PAC7AN	$BVP.DATA.&BASE.AN	Input	Development Database Index

Code	Physical name	Type	Label
PAC7AY	$BVP.DATA.&BASE.AY	Input	Development Database extension data
PAC7JP	$BVP.DATA.&BASE.PJ/G+1	Output	Journal file in a 3.0 format (length=170)
PAC7ET		Report	Report

PJ25 - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '                 RETRIEVAL OF PJ FILE                  ';
COMM '                                                       ';
COMM ' ----------------------------------------------------- ';
COMM '                                                       ';
MVL  OLDPJ='OLDPJ',
     PAC7PJ='$BVP.DATA.'&BASE'.PJ',
     RFTM='DVC=$DVTM,MD=$MDTM';
STEP BVPTU918,FILE=($BVP.SYS.PGM),DUMP=DATA,REPEAT;
     ASG PAC7PJ,&OLDPJ;
     ASG PAC7JP,&PAC7PJ!!/G+1;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AN,NBBUF=10;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=READ,SHARE=MONITOR;
     ASG PAC7AY,$BVP.DATA.&BASE.AY,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AY,NBBUF=4;
     ASG PAC7ET,SYS.OUT;
ESTP;
JUMP ERR,SW20,EQ,1;
SHIFT &PAC7PJ;
```

Procedures - Summary Table of Changes

LIST OF NEW PROCEDURES

Procedure	Comments
ARPM	Archival of the standard bridge journal
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
INQJ	Initialization of the VA Pac interface archive journal and QJ configuration management
VING	Update of Administration Model

Procedure	Comments
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
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

Procedure	Program	Comments
CPSN	PTU850 PTU855	Integrated in PACX procedure
EMSN	PTU810	
MESN	PTU815	
MLIB	PTU100 PTU120	Integrated in PACS procedure
SASN	PTU130 PTU140	Integrated in PACS procedure
SAVE	PTU500	Integrated in PACS procedure
UXSR	UTIXSR	Integrated in PACS procedure
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	

Procedure	Program	Comments
SVAG	PTU550	
GET0	PACTI1	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	
RVKE	PREI40 PREI50	
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 9. Components

Server Environment Components

Introduction

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

- Libraries in which the programs which make up the system, and the parameters required to run the system, are stored:
  - One On-Line Program library,
  - One Batch Program library,
  - One library for the batch procedures' JCLs.
- Permanent files, containing the data handled by the programs defined previously. These files can be classified into two categories:
  - System files, which are not linked to a particular development Database and remain relatively unchanged,
  - Evolving files which are associated to a development Database and whose volumes vary according to the updates performed.

On-Line Documentation

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

Characteristics	Value
Size	around 50000 records for each language
Organization	UIND
Recsize	80
CI Size	4096
Key	12 (position 1)
Utilization	batch and on-line
EFName	$BVP.SYS.AE

Generation Skeletons

The product also includes the following files:

© Copyright IBM Corp. 1983,2003

- A skeleton generation file (SC file) used by the Batch generator function.

Characteristics	Value
Size	Around 45 records
Organization	UIND
Recsize	3204
CI Size	14336
Key	4 (position 1)
EFName	$BVP.SYS.SC

- A skeleton generation file (SG file) used by the On-Line Systems Development and Database generator functions.

Characteristics	Value
Size	Around 450 records
Organization	UIND
Recsize	4605
CI Size	14336
Key	5 (position 1)
Utilization	Batch only
EFName	$BVP.SYS.SG

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

Characteristics	Value
Size	Around 350 records
Organization	UIND
Recsize	4605
CI Size	14336
Key	5 (position 1)
Utilization	Batch only
EFName	$BVP.SYS.SN

- A skeleton generation file (SR file), used by the Reverse generator function.

Characteristics	Value
Size	Around 25 records
Organization	UIND

Characteristics	Value
Recsize	4605
CI Size	14336
Key	5 (position 1)
Utilization	Batch only
EFName	$BVP.SYS.SR

- A skeleton generation file (SP file) used by the PAF function for the generation of extractors:

Characteristics	Value
Size	Around 5 records
Organization	UIND
Recsize	4605
CI Size	14336
Key	5 (position 1)
Utilization	Batch only
EFName	$BVP.SYS.SP

- A skeleton file (SF file) used by the PAF function for the generation of extractors.

Characteristics	Value
Size	Around 3000 records
Organization	Sequential
Recsize	119
Utilization	Batch only
EFName	$BVP.SYS.SF

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

Characteristics	Value
Size	Around 600 records
Organization	UIND
Recsize	4605
CI Size	14336
Key	5 (position 1)

Characteristics	Value
Utilization	Batch only
EFName	$BVP.SYS.SS

Batch Procedure JCL Libraries

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

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

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

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

Administration Database

Administration Database Files

- Data file, GR.
- Extension data file, GY.
- Index file, GN.
- Journal file, GJ.
- User file, GU.
- Journal file of Standard Bridge QJ

Administration Database Backup

The administration Database backup consists of two sequential generation files.

- Backup of the Database (PE).

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

- Backup of the journal (PK).

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

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

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

- Database or Data backup (PC).

This is a sequential backup file of the Development Database components (Data (AR), Index (AN), and extension (AY) if 'Dispatch' option) or of Data only (AR).

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

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

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

Modules - Specific Files

Pac/Impact:

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

DSMS:

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

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

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

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

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

PAF:

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

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

- Work file for PAF in batch mode

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

PAF EXTENSION

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

Complementary Libraries and Files

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

The description of these files is the following one:

PQC FUNCTION

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

- SPECIFIC MEMBERS OF PACBENCH QUALITY CONTROL FUNCTION

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

Chapter 10. Appendix

Installation of the Administration Database Model

VING - Introduction

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

EXECUTION CONDITIONS

The database must be closed to on-line processing.

ABNORMAL EXECUTIONS

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

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

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

VING - Input / Processing / Results

The VING procedure requires two types of user input:

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

The structure of the line is the following:

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

© Copyright IBM Corp. 1983,2003

PRINTED OUTPUT

The procedure outputs:

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

RESULT

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

NOTE

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

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

VING - Description of Steps

UPDATE OF THE ADMINISTRATION DATABASE: VING

Code	Physical name	Type	Label
PAC7AE	$BVP.SYS.AE	Input	error labels
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database Users
PAC7AJ	$BVP.DATA.ADMIN.AJ	Input	Administration Database Journal
PAC7AN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index
PAC7AR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data
PAC7AY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PAC7MA	TMA	Input	IBM Meta-Entities Transactions
PAC7MB	TMBVING	Input	User Input
PAC7BM	TBM	Input/Output	Work file
PAC7WD	TWD	Input/Output	Extracted Transactions
PAC7ES	TES	Input/Output	Extracted Transactions
PAC7TD	TTD	Input/Output	Extracted Transactions

Code	Physical name	Type	Label
PAC7MR	TMR	Output	Extracted Transactions for ROAD
PAC7MX	TMX	Output	Non extracted entities
PAC7IA		Report	General Report of Programs sequence
PAC7EE		Report	Report
PAC7EQ		Report	Report
PAC7EU		Report	Report
PAC7ER		Report	Report
PAC7EZ		Report	Report
PAC7DD		Report	Report
PAC7IE		Report	Report
PAC7IF		Report	Report
PAC7IG		Report	Report
PAC7IH		Report	Report

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.

VING - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '        - DATABASE ADMINISTRATOR                          ';
COMM '        - DICTIONARY UPDATING WITH IBM MODEL ADMIN   -    ';
COMM ' ------------------------------------------------------- ';
COMM '                                                          ';
COMM ' THE VING PROCEDURE PERFORMS A BATCH UPDATE OF THE DATA ';
COMM ' BASE ADMIN. , BASED ON TRANSACTIONS PROVIDED.           ';
COMM '                                                          ';
COMM ' INPUT :                                                 ';
COMM '  -  USER IDENTIFICATION LINE (REQUIRED)                 ';
COMM '     COL  2 : "*"                                        ';
COMM '     COL  3 : USERIDXX                                   ';
COMM '     COL 11 : PASSWORD                                   ';
COMM '     COL 29 : "VINS"                                     ';
COMM '     COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES    ';
COMM ' ------------------------------------------------------- ';
MVL  RFTM='DVC=$DVTM,MD=$MDTM',BASE=ADMIN,
     PAC7EU='DUMMY',PAC7MX='DUMMY',
```

```
     SY=$BVP.SYS.SY,USER=&USER;
IV   ALSY,$BVP.SYS.PROC,
     VL=(SY=&SY,USER=&USER);
CR   IF=*VING,
     OF=(TMBVING,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISIZE=2048,RECSZ=80,RECFORM=FB);
CR   IF=$BVP.SYS.METAD,
     OF=(TMA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(BLKSZ=28600,RECSZ=286,RECFORM=FB);
STEP BVPVINS,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7SY,&SY&USER;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.ADMIN.AJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG H_BJRNL,FILESTAT=TEMPRY,
         DVC=$DVTM,MD=$MDTM;
     ASG PAC7MX,TMX,TEMPRY,END=PASS;
     ASG PAC7EU,SYS.OUT;
     ASG PAC7BM,TBM,TEMPRY,&RFTM;
     ASG PAC7WD,TWD,TEMPRY,&RFTM;
     ASG PAC7MB,TMBVING,TEMPRY,&RFTM;
     ASG PAC7MA,TMA,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ES,TES,TEMPRY,&RFTM;
     ASG PAC7MR,TMR,TEMPRY,&RFTM,END=PASS;
     ASG PAC7TD,TTD,TEMPRY,&RFTM,END=PASS;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7EE,SYS.OUT;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG PAC7IA,SYS.OUT;
     ASG PAC7IG,SYS.OUT;
     ASG PAC7IH,SYS.OUT;
     ASG PAC7EQ,SYS.OUT;
     ASG PAC7ER,SYS.OUT;
     ASG PAC7EZ,SYS.OUT;
     ASG PAC7AR,$BVP.DATA.ADMIN.AR,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.ADMIN.AN,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE;
     ASG PAC7AY,$BVP.DATA.ADMIN.AY,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
```

```
     DEF PACGGU,READLOCK=STAT;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGY,READLOCK=STAT;
ESTP;
JUMP ERR,SW20,EQ,1;
JUMP END;
ERR:
SEND ' VING - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

Installation of the Development Database Model

VINS - Introduction

The VINS procedure performs the batch update of the Development database 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 Development database 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 Development database is ready for either on-line or batch use.

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

NOTE

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

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

VINS - Description of Steps

UPDATE OF THE DEVELOPMENT DATABASE : VINS

Code	Physical name	Type	Label
PAC7AE	$BVP.SYS.AE	Input	Error Labels
PACGGN	$BVP.DATA.ADMIN.AN	Input	Administration Database Index
PACGGR	$BVP.DATA.ADMIN.AR	Input	Administration Database Data
PACGGY	$BVP.DATA.ADMIN.AY	Input	Administration Database Extension
PACGGU	$BVP.DATA.ADMIN.GU	Input	Administration Database Users
PAC7AJ	$BVP.DATA.&BASE.AJ	Input	Development Database Journal
PAC7AN	$BVP.DATA.&BASE.AN	Input	Development Database Index
PAC7AR	$BVP.DATA.&BASE.AR	Input	Development Database Data
PAC7AY	$BVP.DATA.&BASE.AY	Input	Development Database Extension
PAC7MA	TMA	Input	IBM Meta-Entities Transactions
PAC7MB	TMBVINS	Input	User Input
PAC7BM	TBM	Input/Output	Work File
PAC7WD	TWD	Input/Output	Extracted Transactions
PAC7ES	TMES	Input/Output	Extracted Transactions
PAC7TD	TTD	Input/Output	Extracted Transactions
PAC7MR	TMR	Output	Extracted Transactions for REOR
PAC7MX	TMX	Output	Non extracted entities
PAC7IA		Report	Complete printing of programs sequence
PAC7EE		Report	Report
PAC7EQ		Report	Report
PAC7EU		Report	Report
PAC7ER		Report	Report
PAC7EZ		Report	Report
PAC7DD		Report	Report
PAC7IE		Report	Report
PAC7IF		Report	Report
PAC7IG		Report	Report
PAC7IH		Report	Report

Return codes:

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

VINS - Execution JCL

```
COMM 'VisualAge Pacbase 3.0';
COMM '        - DICTIONARY UPDATING WITH IBM MODEL DEVPT   -  ';
COMM '                                                       ';
COMM ' ----------------------------------------------------- ';
COMM '                                                       ';
COMM ' THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE     ';
COMM ' DATABASE, BASED ON TRANSACTIONS PROVIDED.             ';
COMM '                                                       ';
COMM ' INPUT :                                               ';
COMM '  -  USER IDENTIFICATION LINE (REQUIRED)               ';
COMM '     COL  2 : "*"                                      ';
COMM '     COL  3 : USERIDXX                                 ';
COMM '     COL 11 : PASSWORD                                 ';
COMM '     COL 29 : "VINS"                                   ';
COMM '     COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES  ';
COMM '              "R" - RETRIEVAL OF USER ENTITIES WITH THE ';
COMM '                    "EXTENSION" FORMAT                 ';
COMM '              " "   "I" + "R"                          ';
COMM ' ----------------------------------------------------- ';
COMM '                                                       ';
MVL  RFTM='DVC=$DVTM,MD=$MDTM',
     PAC7EU='DUMMY',PAC7MX='DUMMY',
     SY=$BVP.SYS.SY,USER=&USER;
IV   ALSY,$BVP.SYS.PROC,
     VL=(SY=&SY,USER=&USER);
CR   IF=*VINS,
     OF=(TMBVINS,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(CISIZE=2048,RECSZ=80,RECFORM=FB);
CR   IF=$BVP.SYS.METBA,
     OF=(TMA,TEMPRY,&RFTM,END=PASS),
     OUTDEF=(BLKSZ=28600,RECSZ=286,RECFORM=FB);
STEP BVPVINS,FILE=$BVP.SYS.PGM,DUMP=DATA,REPEAT;
     ASG PAC7SY,&SY&USER;
     ASG PAC7AE,$BVP.SYS.AE,
         ACC=READ,SHARE=MONITOR;
     DEF PAC7AE,READLOCK=STAT;
     ASG PAC7AJ,$BVP.DATA.&BASE.AJ,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AJ,JOURNAL=BEFORE;
     ASG H_BJRNL,FILESTAT=TEMPRY,
         DVC=$DVTM,MD=$MDTM;
     ASG PAC7MX,TMX,TEMPRY,END=PASS;
     ASG PAC7EU,SYS.OUT;
     ASG PAC7BM,TBM,TEMPRY,&RFTM;
     ASG PAC7WD,TWD,TEMPRY,&RFTM;
     ASG PAC7MB,TMBVINS,TEMPRY,&RFTM;
```

```
     ASG PAC7MA,TMA,TEMPRY,&RFTM,END=PASS;
     ASG PAC7ES,TES,TEMPRY,&RFTM;
     ASG PAC7MR,TMR,TEMPRY,&RFTM,END=PASS;
     ASG PAC7TD,TTD,TEMPRY,&RFTM,END=PASS;
     ASG PAC7DD,SYS.OUT;
     ASG PAC7EE,SYS.OUT;
     ASG PAC7IE,SYS.OUT;
     ASG PAC7IF,SYS.OUT;
     ASG PAC7IA,SYS.OUT;
     ASG PAC7IG,SYS.OUT;
     ASG PAC7IH,SYS.OUT;
     ASG PAC7EQ,SYS.OUT;
     ASG PAC7ER,SYS.OUT;
     ASG PAC7EZ,SYS.OUT;
     ASG PAC7AR,$BVP.DATA.&BASE.AR,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AR,JOURNAL=BEFORE;
     ASG PAC7AN,$BVP.DATA.&BASE.AN,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AN,JOURNAL=BEFORE;
     ASG PAC7AY,$BVP.DATA.&BASE.AY,
         ACC=WRITE,SHARE=MONITOR;
     DEF PAC7AY,JOURNAL=BEFORE;
     ASG PACGGR,$BVP.DATA.ADMIN.AR,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGR,READLOCK=STAT;
     ASG PACGGN,$BVP.DATA.ADMIN.AN,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGN,READLOCK=STAT;
     ASG PACGGU,$BVP.DATA.ADMIN.GU,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGU,READLOCK=STAT;
     ASG PACGGY,$BVP.DATA.ADMIN.AY,
         ACC=READ,SHARE=MONITOR;
     DEF PACGGY,READLOCK=STAT;
ESTP;
JUMP ERR,SW20,EQ,1;
LMN  SL INFILE=(TMR,TEMPRY,&RFTM),
     LIB=($BVP.DATA.&BASE.USERS),
     COM='MV INFILE:MBREOR_VINS,INFORM=SARF,
     TYPE=DAT,NUMBER=(1,1),REPLACE;';
JUMP END;
ERR:
SEND ' VINS - ABNORMAL END OF RUN  ';
LET  SEV 3;
END:
```

Part Number: DEPD7001301A - 6061

(1P) P/N: DEPD7001301A - 6061