VisualAge Pacbase

Installation Guide Windows 2000 or NT Server & Workstation Components

Version 3.0

Note

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

Second Edition (November 2001)

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.

Contents

Notices

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

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

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

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

Trademarks

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

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

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

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

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

Chapter 1. Foreword

Introduction

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

- Server environment,
- Client environment,
- Communication.

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

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

VisualAge Pacbase Architecture

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

VisualAge Pacbase consists of:

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

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

NOTE: The textual mode remains available for some functionalities.

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

The Server environment

It consists of the following components:

- the system elements: programs, files (on-line help included), and parameters,
- The administror's Database that contains user parameters and other parameters,
- One or more development Database(s).

The Client environment:

It consists of the following components:

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

The communication:

The communication functions enable the Server and the 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,
- VisualAge Java CD-Rom,
- VA Pac documentation CD-Rom.

Bibliography

For information on the communication of controls between the security system installed on site and VisualAge Pacbase (control of user codes, passwords and access authorization), refer to the 'Security System Interfaces' manual.

For information on the procedures used by the Administrator in the following contexts:

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

refer to the 'Administrator's Procedures' manual.

For information on the management of user parameters, (update of access keys, of user codes and access authorizations), refer to the 'Administrator's Workstation: Authorizations and Parameters' manual.

Chapter 2. Prerequisites

Prerequisites for Server Environment

Hardware and Software

- Architecture : a Windows/NT or Windows 2000 server and Windows workstations.
- Processor: workstation with Windows NT server (4.0 minimum) or Windows 2000.
- Memory: 96 Mb. Additional memory may be needed depending on the number of servers installed on the same machine.
- Software:
  - Microsoft Windows Script Host (version 5.1 minimum).
  - Micro Focus Application Server.
- Installation support : CD-ROM drive.

Disk Space

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

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

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

Prerequisites for Client Environment

Hardware

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

- 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: adpated to the site network.
- Memory (RAM): 128 Mb (256 Mb advised).
- Software: Microsoft Windows Script Host (version 5.1 and onwards).

Disk Space

Do not forget to leave sufficient disk space for the product and the installation procedure.

The disk space required to install VisualAge Pacbase Workstation components is:

- 58 Mb for the Administration and Development workbench with its modules,
- 15 Mb for the WorkStation.

Software

The VisualAge Pacbase client components require that a Windows 32 bits be installed on your workstation, ie:

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

Communication

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

Chapter 3. Installation of Server Environment

Installation

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

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

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

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

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

By default, the Server is installed on

C:\Program Files\IBM\VisualAge Pacbase\

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

The installation includes three successive steps:

1. installation of the System, which creates the \SYS\ directory,
2. installation of the administration database, which creates the \DATA\ADMIN\ subdirectory,
3. installation of a development database, which creates the \DATA\[database_name]\ subdirectory.

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

All the databases are created with the installation test data.

System Installation

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

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

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

The installation copies:

- the files for the translation of procedures' labels,
- the skeletons in the various languages of the product,
- the programs
- the procedures
- the database creation utility.

Repository Installation

Administration Database

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

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

The following elements must be specified:

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

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

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

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

Development Databases

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

You do so via the Administrator workbench. (refer to Chapter 'Installation of Client Components', Subchapter 'Administrator/Developer workbench').

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

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

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

You must specify the following elements:

- the Database name (8 char.)
- the Database code (4 char.), as it is declared in the Administration Database.
- the Database port number (betweeen 6000 and 9000)

  Warning: it must be different for each database.
- the Database port number (between 6000 and 9000, unique for each Database).
- the Database language code (Engligh by default)

- the generation Language code (English by default)

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

The following procedures are executed: REST, VINS.

Create New Database

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

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

Chapter 4. Installation/Re-installation of Client Components

Things to Know Before Installing

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

  If Windows Installer is not installed on the workstation, it will be installed automatically.
- You also need Microsoft Windows Script, version 5.0 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.
- By default, the root of all the VisualAge Pacbase Client components is:

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

IMPORTANT NOTE :

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

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

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

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

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

```
C:\Program Files\IBM\VisualAge Pacbase_1\
```

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

```
C:\Program Files\IBM\VisualAge Pacbase_2\
```

Fundamentals of VA Pac Client-Server Communication

FOREWORD: The following lines present the principles of communication between the Client components and the VisualAge Pacbase server. These lines contain important information essential to the choice of communication type and the parameterization of the associated

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

AVAILABLE COMMUNICATION TYPES

If the VisualAge Pacbase server runs on Windows NT, the NT Socket protocol must be used.

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

NOTE: THE MIDDLEWARE

The middleware can be installed :

- On an intermediate machine (remote middleware).

  In this case, the Client components communicate via a gateway (the VisualAge Pacbase Gateway) which must run on a separate machine. This option is not available with the Pacbase Web Connection component.

  NOTE: When the option Communication via a gateway is taken, installation of the middleware must therefore be done separately on the remote machine by selecting the Middleware item in the list of available Client components.

  – When the VisualAge Pacbase Gateway is used, neither the installation nor the configuration of a middleware is required on Developer workstations.
  – This gateway performs a centralized and optimized management of server accesses.
- On each Developer workstation (local middleware).
  – The installation of the middleware on each Developer workstation enables direct communication with the VA PAc Server.
  – Because the middleware is installed on each Developer workstation, it is not necessary to administer a gateway (configure, start, stop).

    NOTE: If the option communication via gateway is not selected, the local installation of the middleware is automatically started during the first installation only -- under a given root -- for one of the following Client components: Administrator & Developer workbench, VisualAge Pacbase WorkStation, eBusiness tools.

    For the Pacbase Web Connection component, the installation of the middleware is made during the first installation of this component -- under a given root -- only if the Context Server is taken.

COMMUNICATION FILES

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

Both files are automatically created, only during the first installation of one of these components under a given root and are located in a directory named 'common'.

NOTE: If the middleware is installed locally on each Developer workstation, you will find both files in the same 'common' directory.

IMPORTANT: To add/delete VisualAge Pacbase databases, or modify parameters related to the communication, you will modify these files directly using a standard editor.

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

The bases.ini file:

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

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

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

The vaplocat.ini file :

The vaplocat.ini file is found on each workstation where the middleware is installed. This file is located in the 'common' sub-directory of the installation root directory.

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

A location :

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

Installation Startup

Insert the installation CD.

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

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

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

Then the list of the VisualAge Pacbase Client components is displayed. Choose the Client component(s) you want to install.

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

Administrator & Developer workbench

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

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

- Administrator workbench

  IMPORTANT NOTE: Installing the Administrator workbench on at least one workstation is required as it will allow for the creation of the VA Pac Database(s), Libraries, 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.
  - Rational Rose Bridge.

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

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

This panel does not appear if VA Pac Databases to be accessed by the workbench have already been defined during a prior installation, under the same root, of Administrator & Developer workbench or of the VA Pac WorkStation.

- If you choose the gateway option, you will only have to enter its IP address.
- If you choose the local middleware option, and if the middleware is not already installed under the same root, the installation script will ask you, later on in the install process, to specify a number of communication parameters.

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

Then, you specify the VA Pac Databases to be accessed by the workbench.

NOTE: This will not be asked if the workbench or the VA Pac WorkStation has already been installed under the same root. In such case, the same VA Pac Databases will be taken into account.

To declare VA Pac Databases, enter a 4-column chart:

1. The Database name, whether already installed or not. Names entered here will be displayed in the connection smartguide, thus showing users which Databases they can connect to.
2. The Database logical code.

   Maximum length: 4 characters.

   If the Database is not installed yet at the server level, please keep in mind this code : it will have to be used then.

In case more than one Database should be operated on your site, you have to declare a Database specifically dedicated to the site administration. Its logical code is the '****' reserved code.

Also, the name given to this Database must be as explicit as possible to make it clearly identifiable among the Databases list displayed in the connection smartguide. Database codes entered here will also be displayed in the connection smartguide.

3. The location name

   Maximum length: 20 characters.

   Default: Location-1

   More than one Database may be associated with one location.

4. In the last column (signOn), you specify the user authentication mode. This indicator has three possible values, to be selected via a combo box.

   VAPac: The user will only have to enter his/her code and password in relation to the VA Pac Database access.

   Middleware: The user will only 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).

   The VA Pac authentication is performed by RACF.

   VAPac Middleware: The user will have to enter a code and password to connect to both the VA Pac Database and the host system.

When you have entered -- if need be -- the communication parameters linked to the local middleware option, the actual installation can then start ; press the [Install] button.

START-UP FILES

The start-up files are :

```
ws_admin.bat
ws_batch.bat
ws_eBusiness.bat
```

These files are to be found in the ADWorkbench directory.

START MENU / PROGRAMS CHOICE

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

```
Administrator-Developer workbench
                                  Administrator
                                  Batch
                                  eBusiness
```

eBusiness Tools

eBUSINESS TOOLS INSTALLATION

The eBusiness Tools are:

- Proxy Generator
- Location Editor

- Services Test Facility
- VisualAge Pacabse Connector

This installation allows the eBusiness Tools to be used independently from Developer workbench, without a connection to the VisualAge Pacbase server. eBusiness tools are installed as VisualAge Java features and tools.

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

You are then asked if you wish to install the middleware locally. However, there will be no such question if the middleware is already present under the installation root.

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

Middleware installed in this context allows communication between generated server and client components.

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

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

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 EBusinessTools directory.

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

MENU START / PROGRAMS CHOICE

Once the installation is over, the workstation 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. 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 Server 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 (local middleware or gateway).

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

This panel does not appear if VA Pac Databases to be accessed by the VA Pac WorkStation have already been defined during a prior installation, under the same root, of the WorkStation or of Administrator & Developer workbench.

- If you choose the gateway option, you will only have to enter its IP address.
- If you choose the local middleware option, and if the middleware is not already installed under the same root, the installation script will ask you, later on in the install process, to specify a number of communication parameters.

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

Then, you specify the VA Pac Databases to be accessed by the WorkStation.

NOTE: This will not be asked if the WorkStation or Administrator & Developer workbench has already been installed under the same root. In such case, the same VA Pac Databases will be taken into account.

To declare VA Pac Databases, enter a 4-column chart:

1. The Database name, whether already installed or not.

   Names entered here will be displayed in the connection box, thus showing users which Databases they can connect to.

   NOTE: You may need to use a parameters personalized 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 at the end of this subchapter.

2. The VA Pac Database logical code.

   Maximum length: 4 characters.

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

   NOTE: This code must be unique in regards to a a given Location (see next item 3.)

3. The location name

   Maximum length: 20 characters.

   Default: Location-1

   More than one Databases may be associated with one location.

4. In the last column (signOn), you specify the user authentication mode. This indicator has three possible values, to be selected via a combo box.

   VAPac: The user will only have to enter his/her code and password in relation to the VA Pac Database access.

   Middleware: The user will only have to enter his/her code and password to connect to the host system The VA Pac authentication is performed by RACF.

   VAPac Middleware: The user wil have to enter a code and password to connect to both the VA Pac Database and the host system.

When you will have entered -- if need be -- the communication parameters linked to the local middleware option, the actual installation can then start ; press the [Install] button.

START-UP FILE The start-up file are :

```
pexec.exe
```

This file is to be found in the SPAC directory.

START MENU / PROGRAMS CHOICE

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

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

Certain 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 repertoire where ″NNN″ indicates the version and ″L″ the language code of the version installed.

The Pacbase.dat file, which necessarily conforms with the most recent installation, is therefore used by default at Station start-up.

However, you can create one or more parameter files. This can be useful in the rare case of many methodologies being installed on one workstation. The change of methodology is allowed at reconnection.

The choice of file name is open but must respect DOS file norms. The extension .dat is recommended.

These DOS files should resemble the Pacbase.dat file and should be stored in the same repertoire.

If the Station is reinstalled, the *.dat files created will not be deleted.

PARAMETER FILE DESCRIPTION

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.

Possible Methodology parameter values 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 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 and context servers.

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

The installer systematically checks the presence of a PERL interpretor.

If no interpreter is detected, a specific directory is created in which you will install it later.

The connection script (cgicgi.pl) is copied in this directory.

When the context server is included in the installation, the middleware is automatically installed except if it is already installed under the same root.

NOTE: Important information relating to communication issues are to be found at the beginning of this chapter.

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

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 commands from the context server setup directory:

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

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

At installation time, the context server is not started 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 context server, enter the following command:

```
pacweb -d
```

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 root used for the installation depends on the installation context. For more details, see the first page of this chapter.

IMPORTANT NOTE:

When the middleware is installed in order to be used with the Administrator & Developer workbench or the VisualAge Pacbase WorkStation and when it is installed for the FIRST time under a given root, the installer asks you to set parameters for the locations defined for the VisualAge Pacbase Databases.

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

- If the middleware is used via the VisualAge Pacbase Gateway, you must define all the locations required for the Databases concerned.

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

  WARNING:

  If there is more than one location to define, either for the same Base or to manage many Bases, 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 the middleware is local, 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.

Additional parameters depend on the the protocol code selected :

- NT 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

  Code Page: Value identifying the coding of characters used by the VA Pac server. 1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MVS CICS CPI-C

  Destination-id entry: BVPSCPI (default value). If you modify this value, it must be the same as the value entered in the Symbolic destination name, a parameter included in the configuration of this communication.

Protocol code page: Value identifying the coding of characters used by the VA Pac server. 1140 (US EBCDIC) or 1146 (UK EBCDIC)

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

Editing communication parameters

The bases.ini file

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

By default this file's access path is:

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

NOTE: A number of parameters in the bases.ini file are not explained here as they involve finer middleware settings, also used by proxies (generated by the eBusiness Tools). These parameters are documented in the Proxy Programming Interface Reference 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].

Each section name from this file will be presented to the user during the connection process: 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 as follows with 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.
  - 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

  WARNING: 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 with information
  - 4 et + : trace for debug
- codePageFile = path and name of the file which contains the conversion table of the code pages.

  Default: ..\middleware\CharConv.txt

PARAMETERS FOR GATEWAY ADAPTER (REMOTE MIDDLEWARE)

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

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

  Default: 5647
- location = location name for the Database

  Maximum length: 20 characters.

  Default : Location-1

  More than one Database can point to the same location.

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

The vaplocat.ini file

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

By default, the path to this file is:

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

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):

- NT SOCKET

```
<LocationName>
COMM_TYPE=SOCKET
MONITOR=BVPSCPI
MESSAGE_LENGTH=8192
IXO_TIMEOUT=30
IXO_ADDRESS=127.0.0.1 3000
```

- MVS CICS SOCKET

```
<LocationName>
COMM_TYPE=TCPMVS
MONITOR=BVPSCPI
MESSAGE_LENGTH=8192
IXO_TIMEOUT=30
HOST_ENCODING=1140 (US) or 1146 (UK)
IXO_ADDRESS=127.0.0.1 3000
IXO_TRANSID=V303
```

- MVS CICS CPI-C

```
<LocationName>
COMM_TYPE=CPIC
MONITOR=BVPSCPI
MESSAGE_LENGTH=8192
IXO_TIMEOUT=30
HOST_ENCODING=1140 (US) or 1146 (UK)
```

DETAILS ON THE PARAMETERS:

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

- COMM_TYPE:

  This parameter identifies the communication protocol in use.

  The possible values are:

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

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

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

- 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_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 8192.
- MONITOR:

  Communication monitor code for VisualAge Pacbase, which is BVPSCPI.

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

Uninstalling Client Components

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

Chapter 5. Tests

List of Utilities

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

Script	Description
ARCH	Archiving of the Administration Database journal
SAVE	Saving of the Administration Database
REOR	Reorganization of the Administration Database
REST	Restoration of the Administration Database
ARCH	Saving of the Development Database journal
SAVE	Saving of the Development Database
MLIB	Library management
REOR	Reorganisation of the Development Database
REST	Restoration of the Development Database
UPDT	Batch update of the Development Database
GPRT	Generation print
PACX exli	Library extraction
PACX expj	Journal extraction
PACX extr	Entities extraction
PACX uxsr	sub-networks extraction

Installation Tests

The VA Pac Installation tests include the following operations:

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

Generation-Print TP and Batch Update Tests

Batch/on-line tests and generation-print tests consist of the following steps:

- On-line use tests:
  - Starting up the test database server.
  - Testing screen branching.
  - Executing some updates.
- Batch updating tests:
  - Executing the UPDT procedure.
- Test on generation and print of programs:
  - Executing the GPRT procedure.

Administration Database Procedures Tests

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

- Archiving of the journal created during the use tests: execute the ARCH procedure, which outputs a PJ(1) file.
- Saving of the Administration Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file.
- Reorganization of the sequential backup, PC(1), of the Administration Database: execute the REOR procedure, which outputs a PC(2) file.
- Restoration of the Administration Database using the PJ(1) archived transaction file and the PC(2) Database Database backup file: execute the REST procedure.

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

Development Database Procedures Tests

The tests on the procedures of the Development Database include the following steps to be executed in this order:

- Archiving of the journal created during the use tests: execute the ARCH procedure, which outputs a PJ(1) file.
- Direct backup of the Development Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file.
- Library management: add/delete a Library in the Development Database: execute the PACS procedure (MLIB option), which outputs a PC(2) file.
- Reorganization of the sequential backup, PC(2), of the Development Database: execute the REOR procedure, which outputs a PC(3) file.
- Restoration of the Development Database using the PJ(1) archived transaction file and the PC(3) Database backup file: execute the REST procedure.

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

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

Extraction-Utility Tests

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

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

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

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

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

Chapter 6. Re-installation of Server

Chapter 7. Retrieval

Retrieval of VisualAge Pacbase 2.0 and 2.5

Operations to be Performed

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

Operations to be performed:

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

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

You must execute the following procedures:

- creation of the Administration Database,
  - INGU: creation and initialization of GU user codes file,
  - REST: initialization of the Administration Database with installation data (enter the access key).
  - VINS: installation of the Administration Model, description at the end of the manual (
- Re-organization of the Administration Database if an end-of-installation message requests it:
  - PACS: backup of the Administration Database (SAVE option),
  - REOR: re-organization of the Administration Database,
  - ARCH: initialization of the Administration Database journal file,
  - REST: restoration of the Administration Database,
- retrieval of the old Database data,
  - PE25: retrieval of user parameters from the PE file which was generated during step 1.

3) Retrieval of a Development Database.

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

You must execute the following procedures:

- PC25: retrieval and re-organization of the old Development Database from the backup of the old Database which was created during step 1.
- 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).

Executing the following procedures is optional but you may choose to do it for optimization sake.

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

Steps 4, 5 and 6 are optional.

4) Retrieval of generation-print commands.

It consists in executing the following procedures:

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

5) Retrieval of PacTransfer parameters (UV).

It consists in executing the following procedure:

- UV25: retrieval of the UV file data.

6) Retrieval of PEI files.

It consists in executing the following procedure:

- PP25: retrieval of the PP file data.

Procedures - Summary Table of Changes

List of new procedures

Procedure	Comments
ARCH	Archiving Administration Database
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
INQJ	Initialization of the VA Pac interface archive journal and QJ configuration management
REOR	Re-organization of Administration Database
REST	Restoration of Administration Database
PACS	Backup of Administration Database
VINS	Update of Administration Model
PACS	Management of development Database
GY25	Retrieval of GY file for UPDP
MB25	Retrieval of MB file for UPDT
GY30	Reprise d'un fichier GY pour UPDP vers 2.5

Procedure	Comments
MB30	Reprise d'un fichier MB pour UPDT vers 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)
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	
SVAG	PTU550	
GET0	PACTI1	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	

Procedure	Program	Comments
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	

Retrieval of User Parameters (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 adminstration 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 2.5 user parameters in the administration database.

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 Adminstrator'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 in which you indicate the characteristics of the development Databases which are to be managed in the new Administration Database. You must specify:

- the Database code: it is the logical code, which will be indicated upon the Database restoration.
- the Database name
- If the Database code or name is not specified, an error message is sent and the procedure cannot be run.

The 'B' line has the following structure:

Position	Length	Value	Meaning
2	1	'B'	Line code
3	4	bbbb	Logical Database name
8	36		Database name

Description of Steps

Processing of User parameters (PE): PTU920

Code		Type	Label
PAC7EN		Input	User parameters, old version
PAC7AE		Input	Error messages
PAC7MB		Input	User input
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU PAC7GY		Input Output	Administration Database users (length=310)
PAC7ET		Report	Error report

Transactions formatting: PAF900

Code		Type	Label
PAC7AR		Input	Administratin Database data
PAC7AN		Input	Administration Database index
PAC7AE		Input	Error labels
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU		Input	Administration Database users
PAC7GY		Input	Update transactions

Code		Type	Label
PAC7MV		Output	Transactions formatting (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME		Output	Working file (length=372)
PAC7MW		Output	Working file (length=170)
PAC7MX		Output	Working file (length=743)
PAC7MY		Output	Working file (length=743)
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		Output	UPDP erroneous transactions (length=310)

Update of the administration Database: PACA15

Code		Type	Label
PAC7AR		Output	Administration Database Data file
PAC7AN		Output	Administration Database Index file
PAC7AY		Output	Administration Database extension
PAC7AJ		Output	Administration Database journal
PAC7AE		Input	Error messages
PACGGN		Input	Administration Database Index file
PACGGR		Input	Administration Database Data file
PACGGY		Input	Administration Database Extension
PACGGU		Input	Administration Database users
PAC7DC		Input	Development Database elements DSMS file
PAC7ME		Input	Work file
PAC7MV		Input	Update transactions
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		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 with no error
- 2 : warning
- 4 : critical error

Execution Script

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

```
REM *
REM * -----------------------------------------------------
REM *
<job id=PE25>

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

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

<script language="VBScript">

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


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


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

Call Msg_Log (Array("1022" , "PACA15"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
WshEnv("PAC7IE") = Rep_USR & "\PE25IEA15.txt"
WshEnv("PAC7IF") = Rep_USR & "\PE25IFA15.txt"
WshEnv("PAC7ME") = Rep_TMP & "\WME.tmp"
WshEnv("PAC7MV") = Rep_TMP & "\WMV.tmp"
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")
```

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

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


</script>
</job>
```

Retrieval of the Development Database (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.

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:

  If in the 2.5 release, 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).

- Speficity 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.

  Caution:

  If there are several type values on the same description in the 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.

  Warning:

  If in the 2.5 release, a line with a type which is not a comment is overriden by a comment line, because of the -G splitting, this override is not transferred to the 3.0 release, it must be done manually in the new database.

Data Structures table type

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

Transformation of U type manuals

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

Long data: users entity, 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.

Warning

If a continuation record is modified in a sub-library, in the 2.5 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 2.5 release.

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

Description of Steps

General processes: PTU911

Code		Type	Label
PAC7MC		Input	Sequential image of the network (old release)
PAC7AE		Input	Error messages
PAC7MB		Input	User input
PAC7PB		Output	Firtst data record (length=153)
PAC7PE		Output	User Entity Occurence definition (2.5 release), Report layouts, and Comments (except the calls of Input Aids) (length=193)
PAC7PG		Ouput	Description of Input Aids and Comments including calls of Input Aids. (length=193)
PAC7PL		Output	Definition and Description of Volumes, Definition and Description of Manuals (length=193)
PAC7PZ		Output	User Entities and description of their Occurences (2.5 release)
PAC7PF		Output	Other records (length=153)
PAC7PM		Output	Report file (length=48)
PAC7ET		Report	Report only if absence of Database code

Manuals and Volumes processing: PTU909

Code		Type	Label
PAC7AE		Input	Error messages
PAC7PB		Input	First data record
PAC7PL		Input	Definition and Description of Volumes and Manuals
PAC7PI		Output	Sorted and re-formatted Volumes Definitions and Descriptions (length=153)
PAC7PM		Input/Output	Report file

Comments processing: PTU92A

Code		Type	Label
PAC7AE		Input	Error messages

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

Meta-Entities processing: PTU912

Code		Type	Label
PAC7AE		Input	Error messages
PAC7PZ		Input	User Entities (2.5 release)
PAC7PB		Input	First data record
PAC7ZP		Output	Development Model records (Definition and Descriptions) (length=193)

Report layout processing: PTU919

Code		Type	Label
PAC7AE		Input	Error messages
PAC7PE		Input	User Entity Occurences Definition (2.5 release), Report layouts and Comments (except calls of input aids)
PAC7PB		Input	First data record
PAC7PH		Input	Description of Input Aids and of their calls in the Comments
PAC7PM		Input/Output	Report file
PAC7ZP		Output	User entity Occurences Definition (2.5 release), Report layouts, and Comments (call of Input Aids included) (length=193)
PAC7PO		Output	Comments (except the call of Input Aids) (length=153)
PAC7PD		Output	First data record (length=153)

User Entities processing: PTU913

Code		Type	Label
PAC7AE		Input	Error messages
PAC7PX		Input	Definition of User Entity Occurences (2.5 release), Report layouts, and Comments (including the call of Input Aids)
PAC7PZ		Input	Definition and Description of the Development Model and Description of User Entity Occurences (2.5 release)
PAC7PB		Input	First data record
PAC7ZP		Output	Long data of the Development Model, User Entities, Report layouts, and Comments (including the calls of Input Aids)

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

Sort of long data: PTU91A

Code		Type	Label
PAC7PZ		Input	Intermediate long data
PAC7ZP		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		Type	Label
PAC7AE		Input	Error messages
PAC7ZP		Input	Sorted long data
PAC7PO		Input	Comments (no call of Input Aids)
PAC7PD		Input	First data record
PAC7PI		Input	Volumes Definition and Description
PAC7PF		Input	Other records
PAC7PM		Input	Report file
PAC7PC		Output	Sequential image of the network (3.0 release)
PAC7ET		Report	Retrieval report

Execution Script

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

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

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

<script language="VBScript">

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


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

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

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

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

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

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

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

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

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

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

Call Msg_Log (Array("1022" , "PTU913"))
'-----------------------------------
```

```
WshEnv("PAC7PB") = Rep_TMP & "\WPD.tmp"
WshEnv("PAC7PD") = Rep_TMP & "\WPB.tmp"
WshEnv("PAC7PX") = Rep_TMP & "\WEP.tmp"
WshEnv("PAC7PZ") = Rep_TMP & "\WZP.tmp"
WshEnv("PAC7ZP") = Rep_TMP & "\WZX.tmp"
Return = WshShell.Run("BVPTU913.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU913")

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

Call Msg_Log (Array("1022" , "PTU91A"))
'-----------------------------------
WshEnv("PAC7PZ") = Rep_TMP & "\WZX.tmp"
WshEnv("PAC7ZP") = Rep_TMP & "\WXZ.tmp"
Return = WshShell.Run("BVPTU91A.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU91A")

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

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

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

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

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

</script>
</job>
```

Generation-Print Commands Retrieval (PG20)

Introduction

Principle

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

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

Condition of execution

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.

Notes

The insertion of update transactions is possible only in libraries or sessions already defined in the Database, otherwise they are rejected. The PG file may contain commands associated with a specific library or session which can be purged later.

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

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

Input - Processing - Results

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

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

The line structure is as follows:

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

Description of Steps

Generation-print commands formatting: PTU908

Code		Type	Label
PAC7IN			Generation-print commands, old release
PAC7OU			Re-formatted generation-print commands (length=150)

Generation-print commands processing: PTU921

Code		Type	Label
PAC7PG		Input	Generation-print commands, old release
PAC7AE		Input	Error labels
PAC7MB		Input	User Entities

Code		Type	Label
PAC7GY		Output	Generation-print commands transactions
PAC7GZ		Output	Script line transactions (length = 310)
PAC7ET		Report	Error report

Transactions formatting: PAF900

Code		Type	Label
PAC7AR		Input	Development Database data
PAC7AN		Input	Development Database index
PAC7AE		Input	Error messages
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU		Input	Administration Database users
PAC7GY		Input	Update transactions
PAC7MV		Output	Formatted transactions (It should be able to contain all input transactions and the elementary deletion transactions which are generated by the multiple deletion transactions) (length=170)
PAC7ME		Output	Working file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		Output	UPDP erroneous transactions (length=310)

Update of the development Database: PACA15

Code		Type	Label
PAC7AR		Output	Development Database Data file
PAC7AN		Output	Development Database index
PAC7AY		Output	Development Database extension
PAC7AJ		Output	Development Database journal
PAC7AE		Input	Error messages
PACGGN		Input	Administration Database Index file
PACGGR		Input	Administration Database Data file
PACGGY		Input	Administration Database extension
PACGGU		Input	Administration Database users
PAC7DC		Input	DSMS file of Development Database Elements
PAC7ME		Input	Working file
PAC7MV		Input	Update transactions
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)

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

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

. Return codes:

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

Transactions formatting: PAF900

Code		Type	Label
PAC7AR		Input	Administratin Database data
PAC7AN		Input	Administration Database index
PAC7AE		Input	Error labels
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU		Input	Administration Database users
PAC7GY		Input	Update transactions
PAC7MV		Output	Transactions formatting (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME		Output	Working file (length=372)
PAC7MW		Output	Working file (length=170)
PAC7MX		Output	Working file (length=743)
PAC7MY		Output	Working file (length=743)
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		Output	UPDP erroneous transactions (length=310)

Update of the administration Database: PACA15

Code		Type	Label
PAC7AR		Output	Administration Database Data file
PAC7AN		Output	Administration Database Index file
PAC7AY		Output	Administration Database extension
PAC7AJ		Output	Administration Database journal
PAC7AE		Input	Error messages
PACGGN		Input	Administration Database Index file
PACGGR		Input	Administration Database Data file
PACGGY		Input	Administration Database Extension
PACGGU		Input	Administration Database users
PAC7DC		Input	Development Database elements DSMS file
PAC7ME		Input	Work file

Code		Type	Label
PAC7MV		Input	Update transactions
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		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 with no error
- 2 : warning
- 4 : critical error

Execution Script

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

<script language="VBScript">
Dim MyProc
MyProc = "PG20"
</script>

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

<script language="VBScript">

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

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


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

Call Msg_Log (Array("1022" , "PTU921"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7GY") = Rep_TMP & "\WGY.tmp"
```

```
WshEnv("PAC7GZ") = Rep_ATMP & "\WGZ.tmp"
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7PG") = Rep_SAVE & "\PG"
WshEnv("PAC7ET") = Rep_USR & "\PG20ET921.txt"
Return = WshShell.Run("BVPTU921.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU921")

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

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

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


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

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

Call Msg_Log (Array("1022", "PAF900"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7GY") = Rep_TMP & "\WGY.tmp"
WshEnv("PAC7ME") = Rep_TMP & "\WME.tmp"
WshEnv("PAC7MV") = Rep_TMP & "\WMV.tmp"
WshEnv("PAC7MW") = Rep_TMP & "\WMW.tmp"
```

```
WshEnv("PAC7MX") = Rep_TMP & "\WMX.tmp"
WshEnv("PAC7MY") = Rep_TMP & "\WMY.tmp"
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

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

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

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


</script>
</job>
```

Generation-Print Commands Retrieval (PG25)

Introduction

Principle

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

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

Execution conditions

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

Printed output

This procedure prints a report on the errors encountered.

Notes

The insertion of update transactions is possible only in libraries or sessions already defined in the Database, otherwise they are rejected. The PG file may contain commands associated with a specific library or session which can be purged later.

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

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

Input - Processing - Results

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

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

The line structure is as follows:

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

Description of Steps

Generation-print commands processing: PTU921

Code		Type	Label
PAC7PG		Input	Generation-print commands, old release
PAC7AE		Input	Error labels
PAC7MB		Input	User Entities
PAC7GY		Output	Generation-print commands transactions
PAC7GZ		Output	Script line transactions (length = 310)
PAC7ET		Report	Error report

Transactions formatting: PAF900

Code		Type	Label
PAC7AR		Input	Development Database data
PAC7AN		Input	Development Database index
PAC7AE		Input	Error messages
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU		Input	Administration Database users
PAC7GY		Input	Update transactions

Code		Type	Label
PAC7MV		Output	Formatted transactions (It should be able to contain all input transactions and the elementary deletion transactions which are generated by the multiple deletion transactions) (length=170)
PAC7ME		Output	Working file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		Output	UPDP erroneous transactions (length=310)

Update of the development Database: PACA15

Code		Type	Label
PAC7AR		Output	Development Database Data file
PAC7AN		Output	Development Database index
PAC7AY		Output	Development Database extension
PAC7AJ		Output	Development Database journal
PAC7AE		Input	Error messages
PACGGN		Input	Administration Database Index file
PACGGR		Input	Administration Database Data file
PACGGY		Input	Administration Database extension
PACGGU		Input	Administration Database users
PAC7DC		Input	DSMS file of Development Database Elements
PAC7ME		Input	Working file
PAC7MV		Input	Update transactions
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		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 without error
- 2 : warning
- 4 : fatal error

Transactions formatting: PAF900

Code		Type	Label
PAC7AR		Input	Administratin Database data
PAC7AN		Input	Administration Database index
PAC7AE		Input	Error labels

Code		Type	Label
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU		Input	Administration Database users
PAC7GY		Input	Update transactions
PAC7MV		Output	Transactions formatting (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME		Output	Working file (length=372)
PAC7MW		Output	Working file (length=170)
PAC7MX		Output	Working file (length=743)
PAC7MY		Output	Working file (length=743)
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		Output	UPDP erroneous transactions (length=310)

Update of the administration Database: PACA15

Code		Type	Label
PAC7AR		Output	Administration Database Data file
PAC7AN		Output	Administration Database Index file
PAC7AY		Output	Administration Database extension
PAC7AJ		Output	Administration Database journal
PAC7AE		Input	Error messages
PACGGN		Input	Administration Database Index file
PACGGR		Input	Administration Database Data file
PACGGY		Input	Administration Database Extension
PACGGU		Input	Administration Database users
PAC7DC		Input	Development Database elements DSMS file
PAC7ME		Input	Work file
PAC7MV		Input	Update transactions
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		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 with no error
- 2 : warning
- 4 : critical error

Execution Script

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

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

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

<script language="VBScript">

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

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

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

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

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

Call Msg_Log (Array("1022" , "PACA15"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
```

```
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
WshEnv("PAC7IE") = Rep_AUSR & "\PG25IE1.A15"
WshEnv("PAC7IF") = Rep_AUSR & "\PG25IF1.A15"
WshEnv("PAC7ME") = Rep_ATMP & "\WME.tmp"
WshEnv("PAC7MV") = Rep_ATMp & "\WMV.tmp"
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15/ADMinistration")


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

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

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

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

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

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


</script>
</job>
```

PEI Retrieval (PP25)

Introduction

Principle

This procedure retrieves the 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.

Notes

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 2.5 Database.

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:

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

Description of Steps

Input recognition: PTU001

Management of production environment : PTU923

Code		Type	Label
PAC7PP		Input	Back up of production environment (old release)
PAC7AE		Input	Error message
PAC7MB		Input	User input
PAC7GY		Output	Records of production environments (length=310)
PAC7ET		Report	Report in case of error

Transactions formatting: PAF900

Code		Type	Label
PAC7AR		Input	Development Database data
PAC7AN		Input	Development Database index
PAC7AE		Input	Error messages
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU		Input	Administration Database users
PAC7GY		Input	Update transactions
PAC7MV		Output	Formatted transactions (It should be able to contain all input transactions and the elementary deletion transactions which are generated by the multiple deletion transactions) (length=170)
PAC7ME		Output	Working file (length=372)
PAC7MW		Output	Work file (length=170)
PAC7MX		Output	Work file (length=743)
PAC7MY		Output	Work file (length=743)
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		Output	UPDP erroneous transactions (length=310)

Update of the development Database: PACA15

Code		Type	Label
PAC7AR		Output	Development Database Data file
PAC7AN		Output	Development Database index
PAC7AY		Output	Development Database extension
PAC7AJ		Output	Development Database journal
PAC7AE		Input	Error messages
PACGGN		Input	Administration Database Index file
PACGGR		Input	Administration Database Data file
PACGGY		Input	Administration Database extension
PACGGU		Input	Administration Database users

Code		Type	Label
PAC7DC		Input	DSMS file of Development Database Elements
PAC7ME		Input	Working file
PAC7MV		Input	Update transactions
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		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 without error
- 2 : warning
- 4 : fatal error

Execution Script

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

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

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

<script language="VBScript">

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

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


Call Msg_Log (Array("1022", "PAF900"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PACGGN") = Rep_BASE & "\AN"
WshEnv("PACGGR") = Rep_BASE & "\AR"
WshEnv("PACGGU") = Rep_BASE & "\GU"
WshEnv("PAC7GY") = Rep_TMP & "\WGY.tmp"
```

```
WshEnv("PAC7ME") = Rep_TMP & "\WME.tmp"
WshEnv("PAC7MV") = Rep_TMP & "\WMV.tmp"
WshEnv("PAC7MW") = Rep_TMP & "\WMW.tmp"
WshEnv("PAC7MX") = Rep_TMP & "\WMX.tmp"
WshEnv("PAC7MY") = Rep_TMP & "\WMY.tmp"
Return = WshShell.Run("BVPAF900.EXE" , 1, TRUE)
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_BASE & "\AN"
WshEnv("PACGGR") = Rep_BASE & "\AR"
WshEnv("PACGGU") = Rep_BASE & "\GU"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
WshEnv("PAC7IE") = Rep_USR & "\PP25IE1A15.txt"
WshEnv("PAC7IF") = Rep_USR & "\PP25IF1A15.txt"
WshEnv("PAC7ME") = Rep_TMP & "\WME.tmp"
WshEnv("PAC7MV") = Rep_TMP & "\WMV.tmp"
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")


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


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


</script>
</job>
```

Retrieval of Pac/Transfer Parameters (UV25)

Introduction

Principle

The UV25 procedure retrieves the UV PacTransfer parameters file, 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.

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

Description of Steps

Input recognition: PTU001

Processing of transfer parameters: PTU922

Code		Type	Label
PAC7UV		Input	Transfer parameters, old release
PAC7AE		Input	Error messages
PAC7MB		Input	User input
PAC7GY		Output	Transfer parameter transactions (length=310)
PAC7ET		Report	Report in case of error

Transactions formatting: PAF900

Code		Type	Label
PAC7AR		Input	Administratin Database data
PAC7AN		Input	Administration Database index
PAC7AE		Input	Error labels
PACGGR		Input	Administration Database data
PACGGN		Input	Administration Database index
PACGGU		Input	Administration Database users
PAC7GY		Input	Update transactions
PAC7MV		Output	Transactions formatting (should be able to contain all input transactions and the elementary cancel transactions generated by multiple cancel transactions) (length=170)
PAC7ME		Output	Working file (length=372)
PAC7MW		Output	Working file (length=170)
PAC7MX		Output	Working file (length=743)
PAC7MY		Output	Working file (length=743)
PAC7RB		Output	UPDT erroneous transactions (length=80)

Code		Type	Label
PAC7RY		Output	UPDP erroneous transactions (length=310)

Update of the administration Database: PACA15

Code		Type	Label
PAC7AR		Output	Administration Database Data file
PAC7AN		Output	Administration Database Index file
PAC7AY		Output	Administration Database extension
PAC7AJ		Output	Administration Database journal
PAC7AE		Input	Error messages
PACGGN		Input	Administration Database Index file
PACGGR		Input	Administration Database Data file
PACGGY		Input	Administration Database Extension
PACGGU		Input	Administration Database users
PAC7DC		Input	Development Database elements DSMS file
PAC7ME		Input	Work file
PAC7MV		Input	Update transactions
PAC7RB		Output	UPDT erroneous transactions (length=80)
PAC7RY		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 with no error
- 2 : warning
- 4 : critical error

Execution Script

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

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

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

<script language="VBScript">
```

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

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

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

Call Msg_Log (Array("1022" , "PACA15"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7DC") = "NUL"
WshEnv("PAC7IE") = Rep_USR & "\UV25IEA15.tmp"
WshEnv("PAC7IF") = Rep_USR & "\UV25IFA15.tmp"
WshEnv("PAC7ME") = Rep_TMP & "\WME.txt"
WshEnv("PAC7MV") = Rep_TMP & "\WMV.txt"
WshEnv("PAC7RB") = "NUL"
WshEnv("PAC7RY") = "NUL"
Return = WshShell.Run("BVPACA15.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PACA15")

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

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


</script>
</job>
```

Retrieval of MB Transactions (MB25)

Introduction

Principle

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

Requisites:

No requisite.

Printed output:

This procedure prints a report of the errors encountered.

Result:

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

Description of Steps

MB file retrieval: PTU926

Code		Type	Label
PAC7AE		Input	Error messages
PAC7AR		Input	Development database data
PACGGR		Input	Administration database data
PACGGN		Input	Administration database index
PACGGU		Input	Administration database users
PAC7MB		Input	2.5 MB transactions
PAC7MV		Output	Retrieval transactions for UPDT
PAC7ME		Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Batch procedure authorization option

Execution Script

```
REM * -----------------------------------------------------
REM *      VISUALAGE PACBASE
REM *
REM * -----------------------------------------------------
REM *                 RETRIEVAL OF MB FILE
REM *
REM * -----------------------------------------------------
REM *
<job id=MB25>

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

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

```
<script language="VBScript">

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


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


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

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


</script>
</job>
```

Retrieval of GY Transactions (GY25)

Introduction

Principle

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

Requisites:

No requisite.

Printed Output:

This procedure prints a report of the errors encountered.

Result:

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

Description of Steps

GY file retrieval: PTU927

Code		Type	Label
PAC7AE		Input	Error messages
PAC7AR		Input	Development database data
PACGGR		Input	Administration database data
PACGGN		Input	Administration database index
PACGGU		Input	Administration database users
PAC7GY		Input	2.5 GY transactions
PAC7MV		Output	Retrieval transactions for UPDP
PAC7ME		Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

Execution Script

```
REM * -----------------------------------------------------
REM *      VISUALAGE PACBASE
REM *
REM * -----------------------------------------------------
REM *                 RETRIEVAL OF GY FILE
REM *
REM * -----------------------------------------------------
REM *
<job id=GY25>

<script language="VBScript">
Dim MyProc
MyProc = "GY25"
</script>

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

<script language="VBScript">

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


Call Msg_Log (Array("1022" , "PTU927"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7GY") = Fic_Input
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7EF") = Rep_USR & "\GY25EF927.txt"
WshEnv("PAC7DD") = Rep_USR & "\GY25DD927.txt"
WshEnv("PAC7ME") = Rep_USR & "\GY25ME.txt"
WshEnv("PAC7MV") = Rep_USR & "\GY25MV.txt"
Return = WshShell.Run("BVPTU927.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU927")


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

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


</script>
</job>
```

Retrieval of MB Transactions (MB30)

Introduction

Principle

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

Requisites:

No requisite.

Printed Output:

This procedure prints a report of the errors encountered.

Result:

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

Description of Steps

MB file retrieval: PTU928

Code		Type	Label
PAC7AE		Input	Error messages
PAC7AR		Input	Development database data
PACGGR		Input	Administration database data
PACGGN		Input	Administration database index
PACGGU		Input	Administration database users
PAC7MB		Input	3.0 MB transactions
PAC7MV		Output	Retrieval transactions for UPDT
PAC7ME		Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

Execution Script

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

```
REM * -----------------------------------------------------
REM *
<job id=MB30>

<script language="VBScript">
Dim MyProc
MyProc = "MB30"
</script>

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

<script language="VBScript">

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


Call Msg_Log (Array("1022" , "PTU928"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7MB") = Fic_Input
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7EF") = Rep_USR & "\MB30EF928.txt"
WshEnv("PAC7DD") = Rep_USR & "\MB30DD928.txt"
WshEnv("PAC7ME") = Rep_USR & "\MB30ME.txt"
WshEnv("PAC7MV") = Rep_USR & "\MB30MV.txt"
Return = WshShell.Run("BVPTU928.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU928")


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

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


</script>
</job>
```

Retrieval of GY Transactions (GY30)

Introduction

Principle

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

Requisites:

No requisite.

Printed Output:

This procedure prints a report of the errors encountered.

Result:

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

Description of Steps

GY file retrieval: PTU929

Code		Type	Label
PAC7AE		Input	Error messages
PAC7AR		Input	Development database data
PACGGR		Input	Administration database data
PACGGN		Input	Administration database index
PACGGU		Input	Administration database users
PAC7GY		Input	3.0 GY transactions
PAC7MV		Output	Retrieval transactions for UPDP
PAC7ME		Output	Revoked transactions
PAC7EF		Report	Retrieval reports
PAC7DD		Report	Authorization option

Execution Script

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

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

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

<script language="VBScript">

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


Call Msg_Log (Array("1022" , "PTU929"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7GY") = Fic_Input
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PAC7EF") = Rep_USR & "\GY30EF929.txt"
WshEnv("PAC7DD") = Rep_USR & "\GY30DD929.txt"
WshEnv("PAC7ME") = Rep_USR & "\GY30ME.txt"
WshEnv("PAC7MV") = Rep_USR & "\GY30MV.txt"
Return = WshShell.Run("BVPTU929.exe" , 1, TRUE)
Call Err_Cod(Return , 0 , "PTU929")
```

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


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


</script>
</job>
```

Chapter 8. Components

Server Environment Components

Introduction

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

- Directories in which the programs which make up the system, and the parameters required to run the system, are stored.
- Permanent files, containing the data handled by the programs defined previously. These files can be classified into two categories:
  - System files, which are not linked to a particular development Database and remain relatively unchanged,
  - Evolving files which are associated to a development Database and whose volumes vary according to the updates performed.

On-Line Documentation

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

Generation Skeletons

The product also includes the following files:

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

Administration Database - Specific Components

Administration Database Files

Administration Database Backup

Development Database

Development Database Files

Development Database Backup Files

Modules - Specific Files

PacImpact:

DSMS:

PAF:

Complementary Libraries and Files

Chapter 9. Appendix

Installation of the Administration Database Model

Introduction

Input - Processing - Results

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

Printed output

The procedure ouputs:

- 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.

IMPORTANT NOTE:

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

Description of Steps

DESCRIPTION OF STEPS

Update of the Administration Database : VINS

Code		Type	Label
PAC7AE		Input	error labels
PACGGN		Input	Administration Database Index
PACGGR		Input	Administration Database Data
PACGGY		Input	Administration Database Extension
PACGGU		Input	Administration Database Users
PAC7AJ		Input	Administration Database Journal
PAC7AN		Input	Administration Database Index

Code		Type	Label
PAC7AR		Input	Administration Database Data
PAC7AY		Input	Administration Database Index
PAC7MA		Input	IBM Meta-Entities Transactions
PAC7MB		Input	User Input
PAC7BM		Input output	User Input
PAC7WD		Input output	Extracted Transactions
PAC7ES		Input output	Extracted Transactions
PAC7TD		Input output	Extracted Transactions
PAC7MR		Output	Extracted Transactions for REOR on Administration Base
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 identified on the files
- 4: Erroneous record of the journal file (Date or session number not numeric)
- 8: No access authorization for batch procedure or invalid database (in such a case, restart the procedure with 'D' in column 16)

Execution Script

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

```
REM *                    WITH THE SAME CODES IN LOWER
REM *                    LEVEL LIBRARIES
REM * ------------------------------------------------------
REM *
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1017 VING
ECHO                  --------------
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1003 %1
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1006 %2
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1018 %3
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1019 %4
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1020 %5
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1021 %6
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1028 %7
ECHO ------------------------------------------------------
ECHO .
CALL %4:%1ÈPROCÈMSGPAUSE %4 %1 %2
ECHO .
CALL %4:%1ÈASSIGNÈ%2ÈPAC7AE
SET  PAC7AJ=%7:%1ÈJOURNALÈ%2ÈAJ
CALL %4:%1ÈASSIGNÈ%2ÈPAC7AN
CALL %4:%1ÈASSIGNÈ%2ÈPAC7AR
CALL %4:%1ÈASSIGNÈ%2ÈPAC7AY
SET  PAC7GJ=%7:%1ÈJOURNALÈ%2ÈGJ
CALL %4:%1ÈASSIGNÈ%2ÈPAC7GN
CALL %4:%1ÈASSIGNÈ%2ÈPAC7GR
CALL %4:%1ÈASSIGNÈ%2ÈPAC7GU
CALL %4:%1ÈASSIGNÈ%2ÈPAC7GY
SET  PAC7MA=%5:%1ÈINPUTÈ%2ÈMAVING
SET  PAC7MB=%5:%1ÈINPUTÈ%2ÈMBVING
SET  PAC7IA=%3ÈVINGIA.IN3
SET  PAC7DD=%3ÈVINGDD.IN3
SET  PAC7EE=%3ÈVINGEE.IN3
SET  PAC7EQ=%3ÈVINGEQ.IN3
SET  PAC7EU=%3ÈVINGEU.IN3
SET  PAC7EZ=%3ÈVINGEZ.IN3
SET  PAC7IE=%3ÈVINGIE.IN3
SET  PAC7IF=%3ÈVINGIF.IN3
SET  PAC7BM=%3ÈBM
SET  PAC7ES=%3ÈES
SET  PAC7WD=%3ÈWD
SET  PAC7MR=%3ÈMR
SET  PAC7TD=%3ÈTD
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1022 VINS
VINS
IF %ERRORLEVEL% NEQ 0 GOTO ERRIN3
REM ------------------------------------------------------
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1023
ECHO .
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1024
DEL  %3ÈBM
DEL  %3ÈES
DEL  %3ÈWD
GOTO END
REM ------------------------------------------------------
:ERRIN3
CALL %4:%1ÈASSIGNÈ%2ÈPACMSG 1025 VINS
PAUSE
:END
ECHO ON
```

Installation of the Development Database Model

Introduction

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

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

EXECUTION CONDITIONS

None.

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.

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:

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

Printed output

The procedure prints:

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

Result

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

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

IMPORTANT NOTE:

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

Description of Steps

DESCRIPTION OF STEPS

Update of the network : VINS

Code		Type	Label
PAC7AE		Input	Error Labels
PACGGN		Input	Administration Database Index
PACGGR		Input	Administration Database Data
PACGGY		Input	Administration Database Extension
PACGGU		Input	Administration Database Users
PAC7AJ		Input	Development Database Journal
PAC7AN		Input	Development Database Index
PAC7AR		Input	Development Database Data
PAC7AY		Input	Development Database Index
PAC7MA		Input	IBM Meta-Entities transactions
PAC7MB		Input	User input
PAC7BM		Input output	User input
PAC7WD		Input output	Extracted transactions
PAC7ES		Input output	Extracted transactions
PAC7TD		Input output	Extracted transactions
PAC7MR		Output	Transactions extracted for REOR
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		Etat	Report
PAC7IG		Etat	Report
PAC7IH		Etat	Report

Return codes:

- 0 : No error detected on files

- 4 : Erroneous record in journal file (Date or session number not numeric)
- 8: No access authorization for batch procedure or invalid database (in such a case, restart the procedure with 'D' in column 16)

Execution Script

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

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

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

<script language="VBScript">

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

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

```
WshEnv("PAC7IF") = Rep_USR & "\VINSIF.txt"
WshEnv("PAC7IG") = Rep_USR & "\VINSIG.txt"
WshEnv("PAC7IH") = Rep_USR & "\VINSIH.txt"


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

WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7MR") = Rep_TMP & "\WMR.tmp"
WshEnv("PAC7TD") = Rep_TMP & "\WTD.tmp"
WshEnv("PAC7WD") = Rep_TMP & "\WWD.tmp"
WshEnv("SYSEXT") = Rep_TMP & "\VINSSYS.tmp"
Return = WshShell.Run("BVPVINS.exe" , 1, TRUE)
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "VINS")

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

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


</script>
</job>
```