VisualAge Pacbase

IBM

Installation Guide
Windows 2000 or NT Server & Client
Components

Version 3.5

VisualAge Pacbase

Installation Guide
Windows 2000 or NT Server & Client Components

Version 3.5

IBM

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.Clément, 93881 Noisy-Le-Grand Cedex. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

Trademarks

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

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

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

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

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

Chapter 1. Foreword

Introduction

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

- Server components
- Client components,
- Communication environment.

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

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

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

VisualAge Pacbase Architecture

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

VisualAge Pacbase consists of:

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

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

NOTE: The textual mode remains available for some functionalities.

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

The Server environment

It consists of the following components:

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

The Client environment

The client environment includes five separately installable components:
- Administrator & Developer workbench (also separately installable).
  Developer workbench includes the following modules (each running independently):
  – Batch,
  – Dialog,
  – eBusiness (includes three eBusiness Tools and Services Modeler),
  – Services Modeler.
  Each of these modules can be executed independently.
- eBusiness Tools:
  – Proxy Generator,
  – Location Editor,
  – Services Test Facility,
  – VisualAge Pacbase Connector,
  – VisualAge Pacbase Web Connection (Dialog Web Revamping Generator),
  – Web application models (WAM).
- VisualAge Pacbase WorkStation (Pacdesign, Pacbench)
- Web application models (WAM)
- Middleware
  The communication functions enable the Server and Client environments to communicate via the main communication protocols on the market.

Contents of Supply

The contents of the supply vary according to the terms of your order:
- Installation Guide,
- CD-Rom or cartridge, depending on the environment, to install the VA Pac server,
- Client Components CD-Rom,
- VA Pac Documentation CD-Rom.

Bibliography

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

- The procedures, used by the Administrator for the Management of Databases and versions, and utilities provided, are documented in the Administrator's procedures Guide.

- For IBM sites, the installation of a Security System, to control user codes, passwords and access authorizations, is documented in the 'Installation of Server Components' chapter, 'Additional Installations' subchapter, and also in the Online Help of Administrator workbench.

Chapter 2. Prerequisites

Prerequisites for Server Environment

Hardware and Software

- Architecture : a Windows/NT, Windows 2000, Windows XP or Windows 2003 server and Windows workstations.
- Processor: workstation with either Windows NT server (4.0 minimum) or Windows 2000 or Windows XP or Windows 2003.
- Memory: 96 Mb. Additional memory may be needed depending on the number of servers installed on the same machine.
- Software:
  - Microsoft Windows Script, version 5.1 or higher. You can download it from the following URL:

    `www.microsoft.com/msdownload/vbscript/scripting.asp`
- Cobol Runtime:
  - For the MICROFOCUS version:

    MICRO FOCUS Application Server
  - For the Acucorp version:

    ACUCOBOL-GT

    CAUTION:

    You must first install the Cobol runtime and update the system environment variables (PATH, COBPATH...).

    For AcuCobol, the PATH system variable must be completed with the path of the AcuCobol runtime (usually ..\AcuGT\bin).
- 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	80
Total for System	74
Total for installation test user files	35

Disk space Installation	Total
Total for installation test Administrator files	260 kb

Prerequisites for Client Environment

Hardware

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

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

Disk Space

Required disk space:

- 38 Mb for Administrator & Developer workbench
- 30 Mb for eBusiness Tools
- 68 Mb for the JDK
- 15 Mb for the Middleware
- 6 Mb for WAM
- 16 Mb for the VisualAge Pacbase WorkStation (with a Methodology included)

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

For an operational installation of AD workbench for example, the minimum required disk space is 121 Mb (38 Mb for AD workbench, 68 Mb for the JDK and 15 for the Middleware).

Software

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

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

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

Communication

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

Chapter 3. Installation of Server Components

Installation

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

- Data1.Cab (any Server file in a compressed format),
- setup.exe (installation executable program),
- setup.ini (installation parameters),
- VisualAge Pacbase Server.Msi (installation table for Windows Installer).

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

By default, the Server is installed on

C:\Program Files\IBM\VisualAge Pacbase\

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

The installation is done in a special directory, \Server, and includes three successive steps:

1. installation of the System, which creates the \SYS\ directory, you must first specify the Generation Language Code (French or English (default language)).
2. installation of the Administration Database, which creates the \CONFIG\ADMIN\ and \DATA\ADMIN\ subdirectories.
3. installation of a Development Database, which creates the \CONFIG\[database_name]\ and \DATA\[database_name]\ subdirectories.

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

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

If the entered code does not correspond to the provided Database (BVAP), neither to a Database defined in the Administration Database, this reference is updated by UPGP in the Administration Database, and you must complete its referencing via Administration workbench.

WARNING:

In this manual, the [database_name] term, used in the presentation of the full directory path, is the name of the directory where are stored the database data. It is not the database code.

System Installation

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

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

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

The installation copies:

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

CAUTION:

During this processing, the error messages file (AE) is updated to take the entered language code into account. An execution report (INAEMSG.log) is created in the WINDOWS temporary directory (\TMP or \TEMP) or directly under C.

Note:

The BVPDS600 and BVPDS610 interface load-modules between VA Pacbase and DSMS 2.5 are provided in standard; for a DSMS use, the DEXP procedure must be adapted to use these new load-modules in place of PDS600 and PDS610 which are not compatible since VA Pacbase 3.0.

An element of the System: the Procedure

Definition

A procedure is a Windows Scripting script (.wsf), which includes Visual Basic Script scripts (.vbs).

Each procedure can be executed only via a launch Script (.wsf) which contains the following information:

- a <resource id> field constituted of input data (141 lines maximum) or the definition of an input file by WshEnv(″BVP_Input″) = ″full file name″ (priority value),
- the user code,
- the Database external name (directory under \DATA),
- the complete assignment of the procedure files that can be parameterized.
  Example: WshEnv(″PACA15_PAC7DC″) = ″DSMS full file name″.

You must enter all this information before launching the procedure.

Other elements can also be specified to assign files or parameterize procedures (e.g. whether the message must be displayed or written in a file).

In any case, the provided user inputs must be verified and adapted to your environment (<resource id> or BVP_Input).

Temporary files, execution reports and output files are found under directories created dynamically.

- Temporary files are located under:
  \DATA\[Base_name]\TMP\[user_code]\[proc_name]-[number]
  They are deleted at the end of the procedure execution. If you don't want these files to be deleted, you must previously set the ″DelDo″ variable to ″NO″ in Init.vbs. So you will have to delete them manually.
  Once this variable is set (″NO″ or ″YES″), it is available for ALL the procedures.
- Output files, execution reports are located under:
  \DATA\[database_name]\USERS\[user_code]\ [proc_name]-[number]
  This number is an application execution number ; by default it is the process number of the procedure.

CAUTION:

In VA Pacbase for Acucorp, when a procedure containing a Cobol sort abends, the temporary sort files may not be deleted. These 16 files are

stored in the WINDOWS temporary directory (\TMP or \TEMP) or directly under 'C:\'. Their date-time correspond to the date and time of the procedure startup.

Execution mode

You can execute the launch script (BVPproc.wsf) or the supervisor (PRBVP.vbs):

- via a command line,
- by double-clicking on them,
- via the 'Start' menu. Access the shortcut: [[dbase_name] Database Utilities] in the [VisualAge Pacbase Server] group of [dbase_name] and enter [Script_name].

NB: The PRBVP.vbs supervisor is an independent VBScript procedure which starts the execution of the launch scripts. It requires 2 arguments: [dbase_name] [script_name] ; [dbase_name] is located under directory \DATA.

To be executed, each procedure requires the definition of various elements:

1. either in the launch script: This launch script, BVPxxxx.wsf, is in Windows Scripting.

   - User code,
   - Database external name (directory under \DATA),
   - The input data is described in the <resource id> field of the script and retrieved in the BVP_Resource environment variable (in this case, the data volume is limited to 141 lines), or can also be contained in a file named in the BVP_Input environment variable (in this case, the data volume is not limited).
   - You can assign files, other than the input file, 'in substitution' by defining an environment variable with:

     . "StepName_FileName" for a substituted assignment in a particular step,

     . or "ProcName_FileName" for a substituted assignment in the whole procedure.

     For example, you define the variable WshEnv("PTU907_PAC7IN") = Fil_In with pathname to indicate a substituted assignment in the PTU907 step of the PPAS procedure.

   - Information or error messages are:

     . either displayed on the screen (MsgTyp = 1),

     . or edited in a file (MsgTyp =2). This is the default value.

     You parameterize this in the BVP_Msg environment variable.

CAUTION: VA Pac does not manage errors and missing elements in the startup script.

1. or in the part which is common to all procedures. This common part, named 'INIT.vbs', is a VBScript inclusion, in each procedure.

   a. Retrieval of the arguments: [base] and [user]

   b. Assignment of 2 parameters with 2 possible values:

      1 DelDo = 'YES'/'NO':

      the temporary files are deleted (YES) or not (NO).

      NoDisp = 'YES'/'NO':

      the Cobol 'display' are not displayed (YES) or are displayed by cmd (NO).

   c. Constitution of the MB file (input data) which contains the user code, password, sometimes a command and analysis elements.

      If the [user] user code is not found in the launch script, it is searched for in the input file.

      If the elements required for the execution of the procedure are absent, an error occurs.

      Note:

      The input file of the retrieval procedures can be defined in the launch script of the procedure via an environment variable: WshEnv("NStep_NFile") = "old 2.5 full file name" which enables the substitution in the procedure. The 'USER' parameter is required for the retrieval procedures, except if the procedure accepts a user input file different from the file of the 2.5 version.

      A PC file (Database backup) is defined for each environment variable: WshEnv("BVP_SaveName").

      Via this variable, the parameter for the definition of assignments (PAC7PC, PAC7PD, PAC7PY) is created. The result PC file is then correctly specified, and the data specific to each option cannot be overwritten.

      If this environment variable is not indicated in the script, the default value is : [Dir_SAVE & "\PC].

      Note: The value of the PC file definition in 2.5 format, in input of the retrieval, can be specified by a substitute value in the launch script: WshEnv("PTU911_PAC7MC"). If this substitute value is not indicated in the script, the default value is: [Dir_SAVE & "\OLDPC"].

Repository Installation

Administration Database

This step creates the Administration Database which monitors the Development 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 Database port number (between 49152 and 65535). This number is unique. The default value is 52029.
- the Database language code (English by default)

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

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

INGU: initialization of the users file

This procedure is executed during the first installation only. It defines and then loads the GU file with the 'TEST' user code (password with blanks) and the 'ADMIN' user code (with the password 'ADMIN'). These user codes will be used to activate the access key and to start tests scripts.

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

Development Databases

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

If the entered code does not correspond to the provided database (BVAP) neither to a database declared in the Administration Database, then this reference is updated by UPGP in the Administration Database.

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

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

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

To install a Development database, you must specify the following parameters:

- the Database name (8 char.)
- the Database code (4 char.), as it is declared in the Administration Database.
- the Database port number (between 49152 and 65535, unique for each Database). The default value, for the first Development Database, is 3676 (reserved number).

  You must specify this number manually in the vaplocat.ini file (see the 'COMMUNICATION FILES' in chapter 4).
- the Database language code (English by default).
- More than one Development Database can be installed. Each has its own environment.

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

The following procedures are executed: UPGP, REST, VINS.

Access key entering

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

- Access key entering,
- Targets definition,
- Key activation.

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

Backup of the Administration Database

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

Create New Database

You can use this icon, in the "Start" menu, to install the Administration Database or a Development Database at any time after the system installation.

The installation requires a user code with the Administrator level.

Users are created in the Administrator workbench. The Administration Database shipped contains a user code with an Administrator level and a password which is "ADMIN".

Configuration parameters

For each Database, a Configuration directory is created with

1- BvpServer.ini, containing the parameters of the conversational server,

2- Fhredir.cfg, containing the "Rollback" parameters.

Unreference a Database

The BVPDELDB procedure is used to unreference a Database on a Windows platform.

This procedure executes the following operations:

- deletion of the emulator service management for a given Database.

- deletion of configuration information (CONFIG directory).

- deletion of information specific to the Database in the Window Registry.

- deletion of shortcuts for the Database.

- renaming of the Database data directories.

Components dates list

INSL procedure

This procedure is used to edit:
- the list of batch, on-line programs and their generation characteristics.
- In case of system operation problem, this list should be printed in order to communicate to VisualAge Pacbase Support all the installation references.

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

INSL - execution Script

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

```
REM *
REM * ------------------------------------------------------
REM *

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

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

' Set Path of programs :
' ----------------------
Reg_PROD = "HKLM\SOFTWARE\IBM\BVP VisualAge Pacbase 3.5\"
Rep_INST =  WshShell.RegRead (Reg_PROD)

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

' Set List of cobol programs :
' ----------------------------
ListCobol = Rep_CFG & "\" & "ListCob.txt"
Set fso = WScript.CreateObject("Scripting.FileSystemObject")
Set dir = fso.GetFolder(Rep_PGM)
Set StreamLst = fso.CreateTextFile(ListCobol, true, false)
Set regEx = New RegExp

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

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

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

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

INSL_LOG=chr(34) & Rep_CFG & "\BVPINSL.log" & chr(34)
INSL_OUT=chr(34) & Rep_CFG & "\BVPINSL.txt" & chr(34)
```

```
WshEnv("PATH") =  Rep_PGM & ";" & WshEnv("PATH")
sCmd = "Bvpdate.exe " & INSL_OUT

If ( CREATE_LOG = True ) Then
  sCmd = "cmd.exe /c " & sCmd & " 1>"  & INSL_LOG & " 2>&1"
  Return = WshShell.Run( sCmd, 7 , TRUE)
Else
  Return = WshShell.Run( sCmd, 10 , TRUE)
End If

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

If ( CREATE_LOG = True ) Then
   Display = Display & vbCRLF & vbCRLF & _
   "File " & INSL_LOG & " has been created."
End If

WScript.Echo(Display)
Wscript.Quit (0)
```

Chapter 4. Installation/Re-installation of Client Components

Things to Know Before Installing

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

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

  http://www.microsoft.com/downloads
- Both Administrator & Developer workbench and eBusiness Tools components require, for their online help, that a 4.7 or higher Netscape version or a 5.5 or higher Internet Explorer version be used.
- The installation of a Client component does not require the prior installation on the server of the VA Pac Database(s) to which it will connect.

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

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

Root Directory

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

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

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

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

NOTE: Do not use blank characters in directory names.

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

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

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

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

Installation Startup

Insert the installation CD-Rom.

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

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

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

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

Choose the Client component you want to install.

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

Fundamentals of VA Pac Client-Server Communication

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

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

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

AVAILABLE COMMUNICATION TYPES

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

MIDDLEWARE

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

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

However you can also choose a communication via a gateway.

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

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

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

COMMUNICATION FILES

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

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

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

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

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

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

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

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

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

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

THE BASES.INI FILE

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

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

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

Administrator & Developer workbench

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

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

- Administrator workbench
- Developer workbench

  Developer workbench includes the following modules, each running independently:
  - Batch module,
  - eBusiness module, also including three of the eBusiness Tools:
    - Proxy Generator,
    - Location Editor,
    - Services Test Facility.
  - Services Modeler module (all of its functionalities being included in the eBusiness module).

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

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

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

This panel does not appear if you have already installed Administrator & Developer workbench or the VA Pac WorkStation under the same root.
- If you choose the direct communication option, the middleware installation script will automatically start once the installation of the workbench is finished. It will ask you to specify a number of communication parameters.

  For complete details on this part of the installation, refer to the Middleware subchapter.
- If you choose the gateway option, enter first the IP address of the gateway here. The installation of the middleware on the Developer workstation - also necessary in this context - will then start automatically after the installation of the workbench.

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

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

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

2. The Database logical code.

    Maximum length: 4 characters.

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

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

3. The location name

    Maximum length: 20 characters.

    Default: Location-1

    More than one Database may be associated with one location.

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

    You select the mode via two check boxes.

    VisualAge Pacbase signon:

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

    Middleware signon:

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

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

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

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

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

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

START-UP FILES

The start-up files are :

- wb_admin.bat
- wb_global.bat
- wb_batch.bat
- wb_dialog.bat
- wb_eBusiness.bat
- wb_services.bat
- wb_extensibility.bat
- wb_cfm.bat

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

START MENU / PROGRAMS CHOICE

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

```
Administrator-Developer workbench
                              Administration
                              Batch
                              Dialog
                              eBusiness
                              Services Modeler
                              Extensibility
                              cfm
                              Global
```

Open Jade and Tidy / Publishing facility

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

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

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

Under 'Self Help' and 'Download', click the 'VisualAge Pacbase downloads' link, then 'VisualAge Pacbase Downloads in English', you will find a number of specific instructions, follow the 'Open Jade' and 'Tidy' links, accordingly.

Open Jade must be installed in the 'adworkbench' directory and Tidy in the 'adworkbench\workstation\publish' directory.

eBusiness Tools

The eBusiness Tools are:

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

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

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

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

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

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

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

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

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

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

START-UP FILES

The start-up files are :

- For the Proxy Generator:

    `vapGen.exe`

- For the Location Editor:

    `vapLocationEditor.exe`

- For the Services Test Facility:

    `vapServicesTestFacility.exe`

- For VisualAge Pacbase Web Connection:

    `PacWebgen.exe`

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

    `ebusinesstools\bin`

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

START MENU / PROGRAMS CHOICE

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

```
eBusinessTools
            Location Editor
            Proxy Generator
            Services test Facility
            Dialog Web Revamping Generator
```

VisualAge Pacbase WorkStation

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

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

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

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

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

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

In the next panel, select the elements to install:

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

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

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

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

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

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

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

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

To do this, a panel enables you to enter:

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

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

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

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

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

2. The Database logical code.

   Maximum length: 4 characters.

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

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

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

3. The location name

   Maximum length: 20 characters.

   Default: Location-1

   More than one Database may be associated with one location.

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

   You select the mode via two check boxes:

   Pacbase is secured:

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

   Middleware is secured:

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

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

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

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

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

START-UP FILE

The start-up file is :

```
pexec.exe
```

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

START MENU / PROGRAMS CHOICE

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

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

INSTALLATION PARAMETERS FILE

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

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

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

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

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

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

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

DESCRIPTION OF THE PARAMETERS FILE

Each of the lines in this file has the following structure:
• a three-digit identifier in positions 1 to 3

- the parameter label, whose position is unfixed
- the parameter value, between brackets ([and]), whose position is also unfixed

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

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

The Pacbase.dat file should not be destroyed.

The possible values for the Methodology parameter are:

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

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

FILES REQUIRED FOR THE USE OF THE GENERATION-PRINT MONITOR

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

This file is saved under the following path:

SPAC\NNNL\GPMON.PRM

If you don't find this file in the directories tree, you must create it in the directories indicated just before. You just have to give it the name GPMON and to enter a .PRM extension.

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

SU bbbb u:/Root/USERS,

bbbb for the Database code,

u for the root driver,

Root for the path to the USERS directory.

NON-STANDARD LABELS FILE

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

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

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

Web Application Models (WAM)

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

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

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

Middleware

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

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

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

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

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

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

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

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

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

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

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

LIST OF PARAMETERS
- VAP SOCKET

  IP address:

  IP address and port used by the VA Pac server.

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

  IP address:

  IP address and port used by the VA Pac server.

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

  Transaction code:

  Code of the CICS transaction of the VA Pac Communication Monitor.

  Code Page:

  Value identifying the coding of characters used by the VA Pac server.

  1140 (US EBCDIC) or 1146 (UK EBCDIC)
- MVS CICS CPI-C

  Destination-id entry:

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

Code page:

Value identifying the coding of characters used by the VA Pac server.

1140 (US EBCDIC) or 1146 (UK EBCDIC)

- MQ-CICS Bridge and MQ-IMS Bridge

  LocCcsid:

  Character set code of the machine in local (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.

  Maximum: 9 characters.

  Queue Manager name:

  Name of the local Queue Manager of the client part. If there is an in-between MQSeries Server on NT, this parameter must identify the in-between NT Queue Manager and not the Queue Manager.

  Maximum: 4 characters.

  Request Queue name:

  Maximum: 48 characters.

  Reply Queue name:

  Maximum: 48 characters.

  Transaction code:

  Maximum of 4 characters for CICS.

  Maximum of 8 characters for IMS.

- MVS IMS Connect

  IP address:

  IP address and port used by the VA Pacbase server.

  Transaction code:

  IMS transaction code of the VA Pacbase Communications Monitor.

  Code page:

  Value identifying the coding of characters used by the VA Pacbase server.

  1140 (US EBCDIC) or 1146 (UK EBCDIC)

  Data Store:

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

  RACF group:

  Name of the RACF group for IMS Connect.

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

Editing Communication Parameters

The bases.ini File

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

By default this file's access path is:
`C:\Program Files\IBM\VisualAge_Pacbase_35\Common\`

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

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

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

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

DESCRIPTION OF A SECTION'S CONTENTS

The parameters in each section are listed below. There one parameter per line:
- baseCode = code of the VisualAge Pacbase Database (required)
  Maximum length: 4 characters

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

- Middleware: indicates that the user will have to give his/her code and password only when he/she connects to the host. The connection to the VA Pac Database will be controlled by RACF (or equivalent).
    - VAPac Middleware: indicates that the user will have to give his/her code and password when he/she connects to the host and to the Database (default option).
- communicationAdapter = indicates the communication mode in use.
    - DIRECT: local middleware
    - GATEWAY: remote middleware (via the VisualAge Pacbase gateway)

The following parameters vary according to the chosen option.
- connectionCleaningInterval = period of time (in millisecond) between two startups of connections cleaning procedures.

    Parameter needed for the VisualAge Pacbase Workstation only.

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

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

PARAMETERS FOR DIRECT ADAPTER (LOCAL MIDDLEWARE)
- locationsFile = indicates the path and name of the file which contains the locations definitions.

    Default: ..\common\vaplocat.ini

    CAUTION: The default value of this parameter should not be modified.
- location = location name for the Database

    Maximum length: 20 characters.

    Default: Location-1

    More than one Database can point to the same location.

    REMINDER: A location identifies the communication protocol used to access the VisualAge Pacbase server and the physical address of this server for this protocol.
- traceFile = path and name of the file which will receive the trace of the middleware execution.

    By default this file is automatically created (with timestamp) in the VapTrace sub-directory.
- traceLevel = trace level of the middleware execution. Its possible values are:
    - 0 : no trace
    - 1 : trace with errors (default)
    - 2 : standard trace, not detailed

- – 3 : trace for information
- – 4 and + : trace for 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.

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

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

  Default: 5647
- location = location name for the Database

  Maximum length: 20 characters.

  Default : Location-1

  More than one Database can point to the same location.

  REMINDER: A location identifies the communication protocol used to access the VisualAge Pacbase server and the physical address of this server for this protocol.
- The traceFile and traceLevel parameters can be used to start the GATEWAY. For more details, refer to the Middleware User's Guide.

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

The vaplocat.ini File

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

By default, the path to this file is:
```
C:\Program Files\IBM\VisualAge_Pacbase_35\Common\
```

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

The maximum length of this name is 20 characters.

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

- VAP SOCKET

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

- MVS CICS SOCKET

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

- MVS CICS CPI-C

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

- MQ-CICS Bridge

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

- IMS Connect

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

- MQ-IMS Bridge

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

DETAILS ON THE PARAMETERS

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

- COMM_TYPE:

  This parameter identifies the communication protocol in use.

  The possible values are:

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

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

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

  MQCICS: VA Pac Server under MVS/CICS, with the use of the MQ-CICS-BRIDGE protocol.

  MQMCICS: Same as MQCICS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.

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

  MQIMS: VA Pac Server under MVS/IMS, with the use of the MQ-IMS-BRIDGE protocol.

  MQMIMS: Same as MQIMS and if the middleware is installed on the same machine as MQSeries Server and you want to use a direct link with MQSeries.

- IXO_ADDRESS:

  IP address and port used by the VA Pac Server.

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

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

  The same for Windows or UNIX, but for these platforms, a number specifically identifies a VA Pac Database. For more details, refer to the

'Installation of Server Environment' chapter, 'Repository Installation' subchapter, 'Development Databases' section, in the respective Installation Guides.

- IXO_DATASTORE:

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

- IXO_DYNAMICREPLYQUEUE

  MQ-CICS Bridge and MQ-IMS Bridge.

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

- IXO_HEADERVERSION

  MQ-CICS Bridge.

  Version of the MQCIH structure: 1 (default) or 2, depending on the used CICS Bridge interface. (1 character maximum).

- IXO_LOCALCCSID

  MQ-CICS Bridge and MQ-IMS Bridge.

  Character set code of the machine in local (819 by default), used for the conversion of messages by MQSeries during the read and write of Queues.

  (9 characters maximum).

- IXO_QUEUEMANAGER

  MQ-CICS Bridge and MQ-IMS Bridge.

  Local Queue Manager name of the client part.

  (4 characters maximum).

  If there is an in-between MQSeries Server on NT, this parameter identifies the in-between NT Queue Manager and not the Queue Manager.

- IXO_RACFGROUP:

  Name of RACF group for IMS Connect.

- IXO_REPLYQUEUE

  MQ-CICS Bridge and MQ-IMS Bridge.

  Name of reply messages Queue.

  (48 characters maximum).

- IXO_REQUESTEXPIRY

  MQ-CICS Bridge and MQ-IMS Bridge.

  Expiration delay of the request message, in seconds.

  (9 characters maximum).

- IXO_REQUESTQUEUE

MQ-CICS Bridge and MQ-IMS Bridge.

Name of request messages Queue.

(48 characters maximum).

- IXO_TIMEOUT:

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

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

- IXO_TRANSID:

Transaction code.

(4 characters maximum for CICS and 8 characters maximum for IMS).

This IMS transaction will have to be declared on IMS by a GEN INPUT IMS as follows:

```
APPLCTN  PSB=BVPSSOC
TRANSACT CODE=WK35,SEGSIZE=32000,MODE=SNGL,SEGNO=00050,
         PRTY=(07,10,00002),PROCLIM=(00005,00015),EDIT=ULC,
         MSGTYPE=(MULTSEG,RESPONSE,$CLS)
```

- HOST_ENCODING:

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

1140 (US EBCDIC) or 1146 (UK EBCDIC)

Not used with MQ-CICS Bridge and MQ-IMS Bridge.

- MESSAGE_LENGTH

The value of this parameter MUST be 31744.

- MONITOR:

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

Not used with IMS Connect and MQ-IMS Bridge.

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

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

Component Updating, Modification, or Removing

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

- 'Update' an already installed component.
- 'Modify' i.e. add a new sub-component to an already installed component.
  This option is available only with:
  - Administrator & Developer workbench:
    To add one of both workbenches as the other is already installed,
  - VisualAge Pacbase WorkStation:
    To add a Methodology.
- 'Remove' an already installed component.
  You can also use the Windows service 'Add/Remove' Programs in the Configuration Panel.

Chapter 5. Tests

List of Utilities

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

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

Installation Tests

The VA Pac Installation tests include the following operations:

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

Generation-Print, Online and Batch Update Tests

These tests consist of the following steps:
- 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

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

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

These tests include the following steps, to be executed in this order:
- Archiving of the journal created during the use tests: execute the ARCH procedure, which outputs a PJ(1) file.
- Backup of the Administration Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file ; as the evaluation key does not allow to execute this procedure, you must enter the previous provided access key.
- Reorganization of the sequential backup, PC(1), of the Administration Database: execute the REOR procedure, which outputs a PC(2) file.
- Restoration of the Administration Database using the PJ(1) archived transaction file and the PC(2) Database backup file: execute the REST procedure.

During all these tests, the Administration Database server must be shut down.
- Extraction of Administration Database data: execute the PAGX procedure and save the GY file of extracted data.
- PAF update of the Administration Database: execute the UPGP procedure with the extracted file (result of the PAGX procedure execution).

Development Database Procedures Tests

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

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

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

- Archiving of the journal created during the use tests: execute the ARCH procedure, which outputs a PJ(1) file.
- Direct backup of the Development Database: execute the PACS procedure (SAVE option), which outputs a PC(1) file ; as the evaluation key does not allow to execute this procedure, you must enter the previous provided access key.
- Library manager: 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).
- Extraction of sub-network: execute the PACS procedure (UXSR option).

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

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

Chapter 6. Re-installation of Server

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

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

Reinstallation

- The system update is automatically executed if you have an older version already installed.

Operational startup scripts, in the \DATA\[Database_name]\SCRIPT directories, are not impacted. To get the last version of the scripts, at re-installation, refer to the startup script models in the \SYS\Script_Models directory.

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

Uninstall the system

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

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

Data deletion

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

The deletion is performed via manual operations:
- Deletion of the directory which contains the Database data, and all its dependent sub-directories,
- Deletion of the directory:
```
Winnt\Profiles\AllUsers\Start Menu\Programs\VisualAge
Pacbase 3.5 Server\[database_name]
```
- Deletion from the HK_Local_Machine register of the key:

```
Software\IBM\BVP VisualAge Pacbase 3.5\Server\BVP_DATA\[database_name].
```

You can automatically perform a complete deletion of Database information (data, registers, services): execute the BVPDELDB procedure (refer to Chapter 3, "Unreference a Database" section).

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

Retrieval of VisualAge Pacbase 2.0 and 2.5

Foreword

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

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

There are two types of procedures:

Procedures to be executed on the 2.5 Database.

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

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

Procedures to be executed on the new Database.

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

These procedures are :
- UTU1 : 'UNS' lines extraction,
- UTU2 : 'UNS' lines update.

Notes

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

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

Example :

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

```
bvptrans F-IN F-OUT 850 923
```

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

Operations to be Performed

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

Screen branching:

It consists of seven steps:

1) Backup of all the old files required. You must execute the following procedures in the old version.
- SAVE: backup of the Development Database (PC),
- PARM: backup of the user parameters (PE),
- SVAG: backup of the generation-print commands (PG)
- SVPE: backup of the PEI environment (PP).

2) Installation of the Administration Database

To install the Administration Database, execute the installation process.

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

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

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

3) Retrieval of a Development Database.

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

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

The execution of the following procedures, even not requested in the VINS report, is advised for a better optimization.
- PACS (SAVE option): backup of the new Development Database,
- REOR: re-organization of new Development Database,
- REST: restoration of the Development Database from the backup file resulting from the preceding re-organization procedure.

Steps 4, 5, 6 and 7 are optional.

4) Retrieval of generation-print commands.

It consists in executing the following procedures:
- PG20: if retrieval of 2.0 PG file,

- PG25: if retrieval of 2.5 PG file.

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

It consists in executing the following procedure:
- UV25: retrieval of the UV file data.

6) Retrieval of PEI files.

It consists in executing the following procedure:
- PP25: retrieval of the PP file data.

7) Retrieval (potential) of PJ journal

It involves executing the PJ25.

Retrieval of User Parameters (PE25)

PE25 - Introduction

Principle

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

Execution conditions

The Administration Database files must be closed to on-line use.

Printed output

This procedure prints a report which indicates the errors encountered.

Result

This procedure integrates the 2.0 or 2.5 user parameters in the Administration Database.

Note

If the Database to be retrieved is under DSMS control, a site-administrator code will must be used at Administration Database connection, and not the ADMIN code provided at installation.

PE25 - Input / Processing / Results

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

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

The 'A' line has the following structure:

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

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

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

The 'B' line has the following structure:

Position	Length	Value	Meaning
2	1	'B'	Line code
3	4	bbbb	Logical Database name
7	36		Database name
42	4	cccc	Transaction code (internal area)

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

The 'C' line has the following structure:

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

PE25 - Description of Steps

Processing of user parameters (PE): PTU920

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

Transaction formatting: PAF900

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

Code	Physical name	Type	Label
PAC7ME	Tmp dir.: WME	Output	Work file (length=372)
PAC7MW	Tmp dir.: WMW	Output	Work file (length=170)
PAC7MX	Tmp dir.: WMX	Output	Work file (length=743)
PAC7MY	Tmp dir.: WMY	Output	Work file (length=743)

Update of the Administration Database: PACA15

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

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

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

PE25 - Execution Script

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

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

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

<script language="VBScript">

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

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

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


Call Msg_Log (Array("1022" , "PTU920"))
'----------------------------------
WshEnv("PAC7MB") = Fic_Input
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU920","PAC7EN",Rep_SAVE & "\OLDPE")
Call BvpEnv("PTU920","PAC7ET",Rep_USR & "\PE25ET920.txt")
Call BvpEnv("PTU920","PAC7GY",Rep_TMP & "\WGY.tmp")
Call RunCmdLog ("BVPTU920")
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("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGY.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Call RunCmdLog ("BVPAF900")
Call Err_Cod(Return , 0 , "PAF900")

Call Msg_Log (Array("1022" , "PACA15"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PE25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PE25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
Call Err_Cod(Return , 0 , "PACA15")

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

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


</script>
</job>
```

Retrieval of the Development Database (PC25)

PC25 - Introduction

Principle

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

Execution conditions

None.

Printed output

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

Result

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

PC25 - Notes on Data Retrieval

Splitting up of the comment description (-G)

The comment description is split up into several descriptions.
- Comments

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

   Caution

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

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

   They include the O line type (-GO).
- Error messages management

   They include the C, D, F ,S ,T, U line types (-GE).
- Call of entities via Relations

They include the R line type (-CR).

- Specificity of the Input Aid entity

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

  WARNING:

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

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

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

Data structures table type

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

Transformation of U type manuals

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

Long data: user entities, input aids, report layouts

There are no more continuation records for these entities. There is only one index for one main record and one index for each continuation record. Long data are 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.

PC25 - Input / Processing / Results

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

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

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

This line must be structured in this way:

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

PC25 - Description of Steps

General processes: PTU911

Code	Physical name	Type	Label
PAC7MC	Save dir. : OLDPC	Input	Sequential image of the network (old release)
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7MB	User input	Input	User input
PAC7PB	Tmp dir. : WPC	Output	First data record (length=153)
PAC7PE	Tmp dir. : WPE	Output	User Entity Occurrence definition (2.5 release), Report layouts, and Comments (except the calls of Input Aids) (length=193)
PAC7PG	Tmp dir. : WPG	Output	Description of Input Aids and Comments including calls of Input Aids. (length=193)
PAC7PL	Tmp dir. : WPL	Output	Definition and Description of Volumes, Definition and Description of Manuals (length=193)
PAC7PZ	Tmp dir. : WPZ	Output	User Entities and description of their Occurrences (2.5 release) (length=193)
PAC7PF	Tmp dir. : WPF	Output	Other records (length=153)
PAC7PM	Tmp dir. : WPM	Output	Report file (length=62)
PAC7ET	User dir. : PC25ET911	Report	Report only if absence of Database code

Manuals and volumes processing: PTU909

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

Comments processing: PTU92A

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

Sort of Input aids : PTU92B

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

Sort of Input aids : PTU92C

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

Report layout processing: PTU919

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

Meta entities processing: PTU912

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

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

User entities processing: PTU913

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

SORT OF LONG DATA: PTU91A

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

Files merging: PTU914

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

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

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

PC25 - Execution Script

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

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

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

<script language="VBScript">

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


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


Call Msg_Log (Array("1022" , "PTU911"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU911","PAC7MC",Rep_SAVE & "\OLDPC")
Call BvpEnv("PTU911","PAC7PB",Rep_TMP & "\WPC.tmp")
Call BvpEnv("PTU911","PAC7PE",Rep_TMP & "\WPE.tmp")
Call BvpEnv("PTU911","PAC7PF",Rep_TMP & "\WPF.tmp")
Call BvpEnv("PTU911","PAC7PG",Rep_TMP & "\WPG.tmp")
```

```
Call BvpEnv("PTU911","PAC7PL",Rep_TMP & "\WPL.tmp")
Call BvpEnv("PTU911","PAC7PZ",Rep_TMP & "\WPZ.tmp")
Call BvpEnv("PTU911","PAC7PM",Rep_TMP & "\WPM.tmp")
Call BvpEnv("PTU911","PAC7ET",Rep_USR & "\PC25ET911.txt")
Call RunCmdLog ("BVPTU911")
Call Err_Cod(Return , 0 , "PTU911")

Call Msg_Log (Array("1022" , "PTU909"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
Call BvpEnv("PTU909","PAC7PB",Rep_TMP & "\WPC.tmp")
Call BvpEnv("PTU909","PAC7PL",Rep_TMP & "\WPL.tmp")
Call BvpEnv("PTU909","PAC7PI",Rep_TMP & "\WPI.tmp")
Call BvpEnv("PTU909","PAC7PM",Rep_TMP & "\WPM.tmp")
Call RunCmdLog ("BVPTU909")
Call Err_Cod(Return , 0 , "PTU909")

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


Call Msg_Log (Array("1022" , "PTU92A"))
'----------------------------------
Call BvpEnv("PTU92A","PAC7PG",Rep_TMP & "\WPG.tmp")
Call BvpEnv("PTU92A","PAC7PE",Rep_TMP & "\WPH.tmp")
Call BvpEnv("PTU92A","PAC7PM",Rep_TMP & "\WPM.tmp")
Call RunCmdLog ("BVPTU92A")
Call Err_Cod(Return , 0 , "PTU92A")

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


Call Msg_Log (Array("1022" , "PTU92B"))
'----------------------------------
Call BvpEnv("PTU92B","PAC7PE",Rep_TMP & "\WPH.tmp")
Call BvpEnv("PTU92B","PAC7PK",Rep_TMP & "\WPK.tmp")
Call RunCmdLog ("BVPTU92B")
Call Err_Cod(Return , 0 , "PTU92B")

Call Msg_Log (Array("1022" , "PTU92C"))
'----------------------------------
Call BvpEnv("PTU92C","PAC7PB",Rep_TMP & "\WPC.tmp")
Call BvpEnv("PTU92C","PAC7PK",Rep_TMP & "\WPK.tmp")
Call BvpEnv("PTU92C","PAC7PD",Rep_TMP & "\WPD.tmp")
Call BvpEnv("PTU92C","PAC7KP",Rep_TMP & "\WKP.tmp")
Call RunCmdLog ("BVPTU92C")
Call Err_Cod(Return , 0 , "PTU92C")

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

```
        Call Msg_Log (Array("1022" , "PTU919"))
        '----------------------------------
        WshEnv("PAC7AE") = Rep_SKEL & "\AE"
        Call BvpEnv("PTU919","PAC7PB",Rep_TMP & "\WPD.tmp")
        Call BvpEnv("PTU919","PAC7PE",Rep_TMP & "\WPE.tmp")
        Call BvpEnv("PTU919","PAC7PM",Rep_TMP & "\WPM.tmp")
        Call BvpEnv("PTU919","PAC7PD",Rep_TMP & "\WPB.tmp")
        Call BvpEnv("PTU919","PAC7PO",Rep_TMP & "\WPO.tmp")
        Call BvpEnv("PTU919","PAC7ZP",Rep_TMP & "\WEP.tmp")
        Call BvpEnv("PTU919","PAC7KP",Rep_TMP & "\WKP.tmp")
        Call RunCmdLog ("BVPTU919")
        Call Err_Cod(Return , 0 , "PTU919")

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

        Call Msg_Log (Array("1022" , "PTU912"))
        '----------------------------------
        WshEnv("PAC7AE") = Rep_SKEL & "\AE"
        Call BvpEnv("PTU912","PAC7PZ",Rep_TMP & "\WPZ.tmp")
        Call BvpEnv("PTU912","PAC7ZP",Rep_TMP & "\WZP.tmp")
        Call BvpEnv("PTU912","PAC7PB",Rep_TMP & "\WPC.tmp")
        Call RunCmdLog ("BVPTU912")
        Call Err_Cod(Return , 0 , "PTU912")

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


        Call Msg_Log (Array("1022" , "PTU913"))
        '----------------------------------
        WshEnv("PAC7AE") = Rep_SKEL & "\AE"
        Call BvpEnv("PTU913","PAC7PB",Rep_TMP & "\WPB.tmp")
        Call BvpEnv("PTU913","PAC7PD",Rep_TMP & "\WPR.tmp")
        Call BvpEnv("PTU913","PAC7PX",Rep_TMP & "\WEP.tmp")
        Call BvpEnv("PTU913","PAC7PZ",Rep_TMP & "\WZP.tmp")
        Call BvpEnv("PTU913","PAC7ZP",Rep_TMP & "\WZX.tmp")
        Call RunCmdLog ("BVPTU913")
        Call Err_Cod(Return , 0 , "PTU913")

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

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

```
Call BvpEnv("PTU91A","PAC7PZ",Rep_TMP & "\WZX.tmp")
Call BvpEnv("PTU91A","PAC7ZP",Rep_TMP & "\WXZ.tmp")
Call RunCmdLog ("BVPTU91A")
Call Err_Cod(Return , 0 , "PTU91A")

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

Call Msg_Log (Array("1022" , "PTU914"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7PC") = BVP_SvName & "-new"
Call BvpEnv("PTU914","PAC7PF",Rep_TMP & "\WPF.tmp")
Call BvpEnv("PTU914","PAC7PI",Rep_TMP & "\WPI.tmp")
Call BvpEnv("PTU914","PAC7PD",Rep_TMP & "\WPR.tmp")
Call BvpEnv("PTU914","PAC7PM",Rep_TMP & "\WPM.tmp")
Call BvpEnv("PTU914","PAC7PO",Rep_TMP & "\WPO.tmp")
Call BvpEnv("PTU914","PAC7ZP",Rep_TMP & "\WXZ.tmp")
Call BvpEnv("PTU914","PAC7ET",Rep_USR & "\PC25ET914.txt")
Call RunCmdLog ("BVPTU914")
Call Err_Cod(Return , 0 , "PTU914")

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

Call Msg_Log (Array("1022" , "BACKUP"))
'-----------------------------------
Call Turnover(BVP_SvName & "")

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

</script>
</job>
```

Generation-Print Commands Retrieval (PG20)

PG20 - Introduction

<u>Principle</u>

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

<u>Execution conditions</u>

The files of the Administration and Development Databases must be closed in the on-line mode.

Printed output

This procedure outputs a report which contains the errors encountered.

Note

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

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

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

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

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

PG20 - Input / Processing / Results

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

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

The line structure is as follows:

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

PG20 - Description of Steps

Generation-print commands formatting: PTU908

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

Generation-print commands processing: PTU921

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

Transaction formatting: PAF900

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

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

Update of the Administration Database: PACA15

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

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

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

Return codes:

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

Transactions formatting: PAF900

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

Update of the Development Database: PACA15

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

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

Return codes:

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

PG20 - Execution Script

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

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

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

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

<script language="VBScript">

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

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

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


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


Call Msg_Log (Array("1022" , "PTU908"))
'-----------------------------------
WshEnv("PAC7IN") = Rep_SAVE & "\PG20"
WshEnv("PAC7OU") = Rep_SAVE & "\OLDPG"
Call RunCmdLog ("BVPTU908")
Call Err_Cod(Return , 0 , "PTU908")

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

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

```
Call Msg_Log (Array("1022" , "PAF900"))
'---------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PAF900","PAC7GY",Rep_TMP & "\WGZ.tmp")
Call BvpEnv("PAF900","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PAF900","PAC7MV",Rep_TMP & "\WMV.tmp")
Call BvpEnv("PAF900","PAC7MW",Rep_TMP & "\WMW.tmp")
Call BvpEnv("PAF900","PAC7MX",Rep_TMP & "\WMX.tmp")
Call BvpEnv("PAF900","PAC7MY",Rep_TMP & "\WMY.tmp")
Call RunCmdLog ("BVPAF900")
If Return <> 0 then
Call Msg_Log (Array("1025", _
              "PAF900/ADMinistration", Return))
End If

If Return = 0 then

Call Msg_Log (Array("1022" , "PACA15"))
'---------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7IE") = Rep_AUSR & "\PG20IE1A15.txt"
WshEnv("PAC7IF") = Rep_AUSR & "\PG20IF1A15.txt"
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
If Return <> 0 then
Call Msg_Log (Array("1025", _
              "PACA15/ADMinistration", Return))
End If

End If
```

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

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

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


Call Msg_Log (Array("1022" , "PACA15"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG20IE2A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG20IF2A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
Call Err_Cod(Return , 0 , "PACA15")
```

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

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


</script>
</job>
```

Generation-Print Commands Retrieval (PG25)

PG25 - Introduction

Principle

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

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

Execution conditions

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

Printed output

This procedure prints a report on the errors encountered.

Note

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

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

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

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

PG25 - Input / Processing / Results

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

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

The line structure is as follows:

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

PG25 - Description of Steps

Generation-print commands processing: PTU921

Code	Physical name	Type	Label
PAC7PG	Save dir. : OLDPG	Input	Generation-print commands, old release
PAC7AE	System - Skel dir. : AE	Input	Error labels
PAC7MB	User input	Input	User Entities
PAC7GY	Tmp dir. : WGY	Output	Generation-print commands transactions (length=310)
PAC7GZ	Tmp dir. : WGZ	Output	Script line transactions (length = 310)
PAC7ET	User dir. : PG25ET921	Report	Error report

Transaction formatting: PAF900

Code	Physical name	Type	Label
PAC7AR	Admin. Base - Base dir.: AR	Input	Administration Database data

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

Update of the Administration Database: PACA15

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

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

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

Return codes:

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

Transactions formatting: PAF900

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

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

Update of the Development Database: PACA15

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

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

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

Return codes:

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

PG25 - Execution Script

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

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

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

<script language="VBScript">

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

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

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

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

```
'Development database
' and <Semadmin> is set to Administration database

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


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

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


If Return = 0 then
Call Msg_Log (Array("1022" , "PACA15"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_ABASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG25IEADM.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG25IFADM.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
```

```
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
Call Err_Cod(Return , 0 , "PACA15/ADMinistration")

End If

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

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

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

Call Msg_Log (Array("1022" , "PACA15"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PG25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PG25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
```

```
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\WDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\WRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\WRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
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 PJ Transactions (PJ25)

PJ25 - Introduction

<u>Principle</u>

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

<u>Execution conditions</u>

None.

<u>Printed output</u>

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

WARNING: The number of written transactions can be very higher than the
read transactions number. From an old 2.n transaction, many
transactions of the new version can be created; in particular for
the meta entities and user entities.

<u>Result</u>

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

Comments

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

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

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

PJ25 - Description of Steps

Processing of PJ transactions sequential file: PTU918

Code	Physical name	Type	Label
PAC7PJ	Save dir.: OLDPJ	Input	journal file old version
PAC7AE	System - skel. dir.: AE	Input	Error messages
PAC7AR	Base dir.: AR	Input	Development Database data
PAC7AN	Base dir.: AN	Input	Development Database index
PAC7AY	Base dir.: AY	Input	Development Database extension data
PAC7JP	Save dir.: PJ	Output	Journal file in the new format (length=170)
PAC7ET	User dir.: PJ25ET918	Report	Report

PJ25 - Execution Script

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

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

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

            <script language="VBScript">

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


            Call Msg_Log (Array("1022" , "PTU918"))
            '----------------------------------
            WshEnv("PAC7AE") = Rep_SKEL & "\AE"
            WshEnv("PAC7AR") = Rep_BASE & "\AR"
            WshEnv("PAC7AN") = Rep_BASE & "\AN"
            WshEnv("PAC7AY") = Rep_BASE & "\AY"
            Call BvpEnv("PTU918","PAC7PJ",Rep_SAVE & "\OLDPJ")
            Call BvpEnv("PTU918","PAC7ET",Rep_USR & "\PJ25ET918.txt")
            Call BvpEnv("PTU918","PAC7JP",Rep_SAVE & "\PJ")
            Call RunCmdLog ("BVPTU918")
            Call Err_Cod(Return , 0 , "PTU918")


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


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


            </script>
            </job>
```

PEI Retrieval (PP25)

PP25 - Introduction

<u>Principle</u>

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

<u>Execution conditions</u>

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

<u>Printed output</u>

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 new format, and updates the Development Database of the installed version.

Note

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

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

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

PP25 - Input / Processing / Results

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

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

The structure of the line is presented as follows:

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

PP25 - Description of Steps

Management of production environment: PTU923

Code	Physical name	Type	Label
PAC7PP	Save dir. : OLDPP	Input	Back up of production environment (old release)

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir. : AE	Input	Error message
PAC7MB	User input	Input	User input
PAC7AR	Base dir. : AR	Input	Development Database data
PAC7AN	Base dir. : AN	Input	Development Database index
PAC7AY	Base dir. : AY	Input	Development Database extension data
PAC7GY	Tmp dir. : WGY	Output	Records of production environments (length=310)
PAC7ET	User dir. : PP25ET923	Report	Report in case of error

Transactions formatting: PAF900

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

Update of the Development Database: PACA15

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

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

Return codes:

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

PP25 - Execution Script

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

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

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

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

<script language="VBScript">

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

Call Msg_Log (Array("1022" , "PTU923"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
Call BvpEnv("PTU923","PAC7GY",Rep_TMP & "\WGY.tmp")
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTU923","PAC7PP",Rep_SAVE & "\OLDPP")
Call BvpEnv("PTU923","PAC7ET",Rep_USR & "\PP25ET923.txt")
Call RunCmdLog ("BVPTU923")
Call Err_Cod(Return , 0 , "PTU923")


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

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

```
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\PP25IE1A15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\PP25IF1A15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
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 passwords (UTMP)

UTMP - Introduction

<u>Principle</u>

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

<u>Execution conditions</u>

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

<u>Printed report</u>

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

Result

The procedure integrates 2.0 or 2.5 users passwords in the Administration Database.

UTMP - Description of Steps

2.N passwords retrieval: PTURMP

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

UTMP - Execution Script

```
REM * ------------------------------------------------------
REM *         VISUALAGE PACBASE
REM *
REM * ------------------------------------------------------
REM *                    PASSWORD RETRIEVAL
REM *
REM * ------------------------------------------------------
REM *
<job id=UTMP>

<script language="VBScript">
Dim MyProc
MyProc = "UTMP"
</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" , "PTURMP"))
'---------------------------------
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTURMP","PAC7CE",Rep_SAVE & "\OLDPE")
Call RunCmdLog ("BVPTURMP")
Call Err_Cod(Return , 0 , "PTURMP")

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

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

Retrieval of Pac/Transfer Parameters (UV25)

UV25 - Introduction

Principle

The UV25 procedure retrieves the UV PacTransfer parameters file, 2.0 or 2.5 release, in the new format.

It updates the Administration Database.

Execution conditions

The Administration Database files must be closed to on-line use.

Printed output

This procedure prints a report on the errors encountered.

UV25 - Input / Processing / Results

A '*' line with a user code, a password and the code of the Development Database concerned by the transfers.

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

The line structure must be as follows:

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

UV25 - Description of Steps

Processing of transfer parameters: PTU922

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

Transaction formatting: PAF900

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

Update of the Administration Database: PACA15

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

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

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

UV25 - Execution Script

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

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

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

<script language="VBScript">

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

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

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

Call Msg_Log (Array("1022" , "PACA15"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_AJOURNAL & "\AJ"
WshEnv("PAC7AN") = Rep_ABASE & "\AN"
WshEnv("PAC7AR") = Rep_ABASE & "\AR"
```

```
WshEnv("PAC7AY") = Rep_ABASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
Call BvpEnv("PACA15","PAC7IE",Rep_USR & "\UV25IEA15.txt")
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\UV25IFA15.txt")
Call BvpEnv("PACA15","PAC7ME",Rep_TMP & "\WME.tmp")
Call BvpEnv("PACA15","PAC7MV",Rep_TMP & "\WMV.tmp")
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
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>
```

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

MB25 - Introduction

<u>Principle</u>

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

<u>Limits</u>

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

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

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

MB25 - Description of Steps

MB file retrieval: PTU926

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

MB25 - Execution Script

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

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

```
        </script>

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

        <script language="VBScript">

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


        Call Msg_Log (Array("1022" , "PTU926"))
        '----------------------------------
        WshEnv("PAC7AE") = Rep_SKEL & "\AE"
        WshEnv("PAC7AR") = Rep_BASE & "\AR"
        Call BvpEnv("PTU926","PAC7MB",RepT_USR & "\OLDMB")
        WshEnv("PACGGN") = Rep_ABASE & "\AN"
        WshEnv("PACGGR") = Rep_ABASE & "\AR"
        WshEnv("PACGGU") = Rep_ABASE & "\GU"
        Call BvpEnv("PTU926","PAC7EF",Rep_USR & "\MB25EF926.txt")
        Call BvpEnv("PTU926","PAC7DD",Rep_USR & "\MB25DD926.txt")
        Call BvpEnv("PTU926","PAC7ME",Rep_USR & "\MB25ME.txt")
        Call BvpEnv("PTU926","PAC7MV",Rep_USR & "\MB25MV.txt")
        Call RunCmdLog ("BVPTU926")
        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>
```

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

GY25 - Introduction

<u>Principle</u>

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

<u>Limits</u>

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

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

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

GY25 - Description of Steps

GY file retrieval: PTU927

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

GY25 - Execution Script

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

```
REM * ---------------------------------------------------------
REM *
<job id=GY25>

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

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

<script language="VBScript">

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


Call Msg_Log (Array("1022" , "PTU927"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU927","PAC7GY",RepT_USR & "\OLDGY")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU927","PAC7EF",Rep_USR & "\GY25EF927.txt")
Call BvpEnv("PTU927","PAC7DD",Rep_USR & "\GY25DD927.txt")
Call BvpEnv("PTU927","PAC7ME",Rep_USR & "\GY25ME.txt")
Call BvpEnv("PTU927","PAC7MV",Rep_USR & "\GY25MV.txt")
Call RunCmdLog ("BVPTU927")

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>
```

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

MB30 - Introduction

<u>Principle</u>

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

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

MB30 - Description of Steps

MB file retrieval: PTU928

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AR	Base dir. : AR	Input	Development Database data
PACGGR	Admin Base - Base dir. : AR	Input	Administration Database data
PACGGN	Admin Base - Base dir. : AN	Input	Administration Database index
PACGGU	Admin Base - Base dir. : GU	Input	Administration Database users
PAC7MB	User dir.: MB30	Input	MB transactions of the new version
PAC7MV	User dir. : MB30MV	Output	Retrieval transactions for UPDT
PAC7ME	User dir. : MB30ME	Output	Revoked transactions
PAC7EF	User dir. : MB30EF928	Report	Retrieval reports
PAC7DD	User dir. : MB30DD928	Report	Authorization option

MB30 - Execution Script

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

```
REM *
<job id=MB30>

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

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

<script language="VBScript">

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


Call Msg_Log (Array("1022" , "PTU928"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU928","PAC7MB",RepT_USR & "\MB30.txt")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU928","PAC7EF",Rep_USR & "\MB30EF928.txt")
Call BvpEnv("PTU928","PAC7DD",Rep_USR & "\MB30DD928.txt")
Call BvpEnv("PTU928","PAC7ME",Rep_USR & "\MB30ME.txt")
Call BvpEnv("PTU928","PAC7MV",Rep_USR & "\MB30MV.txt")
Call RunCmdLog ("BVPTU928")
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>
```

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

GY30 - Introduction

<u>Principle</u>

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

<u>Limits</u>

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

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

Execution conditions

None.

Printed output

This procedure prints a report of the errors encountered.

Result

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

GY30 - Description of Steps

GY file retrieval: PTU929

Code	Physical name	Type	Label
PAC7AE	System - Skel dir. : AE	Input	Error messages
PAC7AR	Base dir. : AR	Input	Development Database data
PACGGR	Admin Base - Base dir. : AR	Input	Administration Database data
PACGGN	Admin Base - Base dir. : AN	Input	Administration Database index
PACGGU	Admin Base - Base dir. : GU	Input	Administration Database users
PAC7GY	User dir. : GY30	Input	GY transactions of the new version
PAC7MV	User dir. : GY30MV	Output	Retrieval transactions for UPDP
PAC7ME	User dir. : GY30ME	Output	Revoked transactions
PAC7EF	User dir. : GY30EF929	Report	Retrieval reports
PAC7DD	User dir. : GY30DD929	Report	Authorization option

GY30 - Execution Script

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

```
REM * ----------------------------------------------------------
REM *
<job id=GY30>

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

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

<script language="VBScript">

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


Call Msg_Log (Array("1022" , "PTU929"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTU929","PAC7GY",RepT_USR & "\GY30.txt")
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
Call BvpEnv("PTU929","PAC7EF",Rep_USR & "\GY30EF929.txt")
Call BvpEnv("PTU929","PAC7DD",Rep_USR & "\GY30DD929.txt")
Call BvpEnv("PTU929","PAC7ME",Rep_USR & "\GY30ME.txt")
Call BvpEnv("PTU929","PAC7MV",Rep_USR & "\GY30MV.txt")
Call RunCmdLog ("BVPTU929")

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>
```

Procedures - Summary Table of Changes

List of new procedures since the 2.5

Procedure	Comments
ARCH	Archiving Administration Database
ARPM	Archiving the QJ journal of SCM module

Procedure	Comments
INAE	Initialization of the error messages file (AE)
INGU	Initialization of the user codes files (GU)
INQJ	Initialization of SCM Tools Interface archive journal
GPRC	Generation (COBOL API)
REOR	Re-organization of Administration Database
REST	Restoration of Administration Database
PACS	Backup of Administration Database
PAGX	Extraction of Administration Database
VINS	Update of Administration Model
PACS	Management of Development Database
MLIB	Management of libraries
SAVE	Backup of Development Database
GY25	Retrieval of GY file for UPDP to 3.n
MB25	Retrieval of MB file for UPDT to 3.n
GY30	Retrieval of GY file for UPDP to 2.n
MB30	Retrieval of MB file for UPDT to 2.n
PC25	Retrieval of Development Database
PE25	Retrieval of user parameters (PE)
PG20	Retrieval of 2.0 generation-print commands (PG)
PG25	Retrieval of 2.5 generation-print commands (PG)
PJ25	Retrieval of journal 2.n
PP25	Retrieval of PEI environment (PP)
UTMP	Retrieval of passwords
UV25	Retrieval of PacTransfer parameters (UV)
UTM1	Conversion of meta entities (step 1)
UTM2	Conversion of meta entities (step 2)
UTU1	Extract line 'UNS' 3.0 utility
UTU2	Update line 'UNS' 3.0 utility'
CHPM	Environment and Database elements check report
CPPM	Comparison between Database and user configuration
EXPM	Extraction of environments
GPPM	Generation report written into QJ
HIPM	Implementation of elements

Procedure	Comments
SIPM	Generation simulation
UPPM	Update of elements
TRED	PacTransfer: print of parameters
UPGP	PAF update Administration Database

List of procedures suppressed since 2.5 release

Procedure	Program	Comments
CPSN	PTU850 PTU855	Integrated in PACX procedure
EMSN	PTU810	
MESN	PTU815	
SASN	PTU130 PTU140	Integrated in PACS procedure
UXSR	UTIXSR	Integrated in PACS procedure
CRYP	PACU99	
PARM	PACU15 PACU80	
LOAE	PACU80	
REAG	PTU560	
SVAG	PTU550	
GET0	PACTI1	
GET1	PACT41	
GET2	PACT41 PACT51	
GRPE	PACR40	
INPE	PACR01	
PP16	PACR90	
PRPE	PACR10	
RSPE	PACR61	
SVPE	PACR60	
RVDE	PREI00 PRE986	
RVKE	PREI40 PREI50	
STOP	PTUR00	
TRRT	REUV802 PTUG90	
VDWN	PVA100 PVA110	
VPUR	PVA400	
VPU1	PVA300 PVA305 PVA310	

Procedure	Program	Comments
VPU2	PVA320	
LVBL	PTULVB	
QREO	PTUN00 PTUN10 PTUN40	
RPPG	PTU908	
RPTD	PTAR20	

Retrieval of VisualAge Pacbase 3.0

Operations to be Performed

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

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

- If this version is installed in the same environment than the 3.0, refer to the 'Re-installation of Server' chapter in this manual.

  After the Administration Model update (VINS execution), connect to the Administrator workbench to enter the new access key.

- If this version is installed in a new environment, after the execution of all the steps of the server part installation, the Databases must be restored following the steps:

  – REST : restoration of the Administration Database,

  – VINS : update of the Administration Model,

  – Connection via the Administration workbench to enter and activate the new access key to the system,

  – REST : restoration of each Development Database,

  – VINS : update of the Development Model for each Development Database.

- In any case, the reorganization of Databases is required (ARCH, SAVE, REOR, REST).

Chapter 8. Components

Server Environment Components

Introduction

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

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

On-Line Documentation

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

Generation Skeletons

The product also includes the following files:

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

Administration Database

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

This file is sensitive and its backup must be executed in a secure environment of the installation site.
- Journal file of SCM Tools Interface, QJ.

Administration Database Backup

The Administration Database backup consists of two sequential generation files.
- Backup of the Database (PC).

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

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

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

The backup of QJ is the JQ file.

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

Development Database

Development Database Files

These files contain all data related to applications development.
- The Data file (AR).

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

All the transactions performed on the Database in batch or on-line mode are saved for two reasons:
- To allow Database restoration if the system standard securities were to fail.
- This information may be used for statistical purposes.

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

Development Database Backup Files

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

This is a sequential backup file of the Development Database components (Data (AR), Index (AN), and extension (AY)).
- The backup of the Development Database index (PD).
- The backup of the Development Database random data (PY).
- Journal backup (PJ)

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

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

Modules - Specific Files

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

DSMS:

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

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

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

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

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

PAF:

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

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

- Work file for PAF in batch mode

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

PAF extension

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

Complementary Libraries and Files

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

PQC function

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

- Specific members of Pacbench Quality Control function

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

Chapter 9. Appendix

Installation of the Administration Database Model

VINS - Introduction

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

Execution conditions

The Database must be closed to on-line processing.

Abnormal execution

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

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

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

VINS - Input / Processing / Results

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

The structure of the line is the following:

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

Position	Length	Value	Meaning
33	1	'I'	Installation of IBM Meta Entities

Printed output

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

Result

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

Note

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

VINS - Description of Steps

Update of the Administration Database: VINS

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

Code	Physical name	Type	Label
PAC7MA	System - skel. dir.: BVPMETAD	Input	IBM Meta Entities Transactions
PAC7MB	User input	Input	User Input
PAC7BM	Tmp dir.: WBM	Input/Output	Work file
PAC7WD	Tmp dir.: WWD	Input/Output	Extracted Transactions
PAC7ES	Tmp dir.: WES	Input/Output	Extracted Transactions
PAC7TD	User dir.: TD	Input/Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR on Administration Base
PAC7MX	NUL	Output	Non extracted entities
PAC7IA	User dir.: VINSIA	Report	General Report of Programs sequence
PAC7EE	NUL	Report	Report
PAC7EQ	User dir.: VINSEQ	Report	Report
PAC7EU	User dir.: VINSEU	Report	Report
PAC7ER	User dir.: VINSER	Report	Report
PAC7EZ	User dir.: VINSEZ	Report	Report
PAC7DD	User dir.: VINSDD	Report	Report
PAC7IE	User dir.: VINSIE	Report	Report
PAC7IF	User dir.: VINSIF	Report	Report
PAC7IG	User dir.: VINSIG	Report	Report
PAC7IH	User dir.: VINSIH	Report	Report

Return codes:

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

VINS - Execution Script

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

```
REM *          - DICTIONARY UPDATING WITH IBM MODEL DEVPT    -
REM *
REM * --------------------------------------------------------
REM *
REM * THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
REM * DATABASE, BASED ON TRANSACTIONS PROVIDED.
REM *
REM * INPUT :
REM *  -  USER IDENTIFICATION LINE (REQUIRED)
REM *     COL  2 : "*"
REM *     COL  3 : USERIDXX
REM *     COL 11 : PASSWORD
REM *     COL 29 : "VINS"
REM *     COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES
REM *              "R" - RETRIEVAL OF USER ENTITIES WITH THE
REM *                    "EXTENSION" FORMAT
REM *              " "   "I" + "R"
REM * --------------------------------------------------------
REM *
<job id=VINS>

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

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

<script language="VBScript">

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

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

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


Call Msg_Log (Array("1022" , "VINS"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("VINS","PAC7BM",Rep_TMP & "\WBM.tmp")
Call BvpEnv("VINS","PAC7ES",Rep_TMP & "\WES.tmp")
Call BvpEnv("VINS","PAC7DD",Rep_USR & "\VINSDD.txt")
WshEnv("PAC7EE") = Rep_TMP & "\NULEE.tmp"
'PAC7EE not used, on default
```

```
            Call BvpEnv("VINS","PAC7EQ",Rep_USR & "\VINSEQ.txt")
            Call BvpEnv("VINS","PAC7ER",Rep_USR & "\VINSER.txt")
            Call BvpEnv("VINS","PAC7EU",Rep_USR & "\VINSEU.txt")
            Call BvpEnv("VINS","PAC7EZ",Rep_USR & "\VINSEZ.txt")
            Call BvpEnv("VINS","PAC7IA",Rep_USR & "\VINSIA.txt")
            Call BvpEnv("VINS","PAC7IE",Rep_USR & "\VINSIE.txt")
            Call BvpEnv("VINS","PAC7IF",Rep_USR & "\VINSIF.txt")
            Call BvpEnv("VINS","PAC7IG",Rep_USR & "\VINSIG.txt")
            Call BvpEnv("VINS","PAC7IH",Rep_USR & "\VINSIH.txt")


            If base = "ADMIN"  Then
            Call BvpEnv("VINS","PAC7MA",Rep_SKEL & "\BVPMETAD")
            Else
            Call BvpEnv("VINS","PAC7MA",Rep_SKEL & "\BVPMETBA")
            End If

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

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

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

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


            </script>
            </job>
```

Installation of the Development Database Model

VINS - Introduction

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

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

Execution conditions

The Database must be closed to on-line processing.

Abnormal execution

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

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

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

VINS - Input / Processing / Results

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

The structure of the line is the following:

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

Printed output

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

<u>Result</u>

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

For the conversion of the user entities into extension format, a sequential file of purge transactions can be generated (if 'R' in column 33 in the user input). You then have to reorganize the Database with this file in input. The reorganization of the Database with the PC file, saved after the retrieval, is also required if the client user entities of the meta entities exist in several sessions.

<u>Note</u>

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

VINS - Description of Steps

Update of the Development Database : VINS

Code	Physical name	Type	Label
PAC7AE	System - Skel. dir.: AE	Input	Error Labels
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database Index
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database Data
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database Extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database Users
PAC7AJ	Journal dir.: AJ	Output	Development Database Journal
PAC7AN	Base dir.: AN	Output	Development Database Index
PAC7AR	Base dir.: AR	Output	Development Database Data
PAC7AY	Base dir.: AY	Output	Development Database Extension

Code	Physical name	Type	Label
PAC7MA	System - skel. dir.: BVPMETBA	Input	IBM Meta Entities Transactions
PAC7MB	User input	Input	User Input
PAC7BM	Tmp dir.: WBM	Input/ Output	User input
PAC7WD	Tmp dir.: WWD	Input/ Output	Extracted Transactions
PAC7ES	Tmp dir.: WES	Input/ Output	Extracted Transactions
PAC7TD	User dir.: TD	Input/ Output	Extracted Transactions
PAC7MR	Tmp dir.: WMR	Output	Extracted Transactions for REOR
PAC7MX	NUL	Output	Non extracted entities
PAC7IA	User dir.: VINSIA	Report	Complete printing of programs sequence
PAC7EE	NUL	Report	Report
PAC7EQ	User dir.: VINSEQ	Report	Report
PAC7EU	User dir.: VINSEU	Report	Report
PAC7ER	User dir.: VINSER	Report	Report
PAC7EZ	User dir.: VINSEZ	Report	Report
PAC7DD	User dir.: VINSDD	Report	Report
PAC7IE	User dir.: VINSIE	Report	Report
PAC7IF	User dir.: VINSIF	Report	Report
PAC7IG	User dir.: VINSIG	Report	Report
PAC7IH	User dir.: VINSIH	Report	Report

Return codes:

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

VINS - Execution Script

```
REM * -------------------------------------------------------
REM *        VISUALAGE PACBASE
REM *
REM * -------------------------------------------------------
REM *        - DICTIONARY UPDATING WITH IBM MODEL DEVPT   -
```

```
REM *
REM * ------------------------------------------------------
REM *
REM * THE VINS PROCEDURE PERFORMS A BATCH UPDATE OF THE
REM * DATABASE, BASED ON TRANSACTIONS PROVIDED.
REM *
REM * INPUT :
REM *  -  USER IDENTIFICATION LINE (REQUIRED)
REM *     COL  2 : "*"
REM *     COL  3 : USERIDXX
REM *     COL 11 : PASSWORD
REM *     COL 29 : "VINS"
REM *     COL 33 : "I" - INSTALLATION OF IBM META-ENTITIES
REM *              "R" - RETRIEVAL OF USER ENTITIES WITH THE
REM *                    "EXTENSION" FORMAT
REM *              " "   "I" + "R"
REM * ------------------------------------------------------
REM *
<job id=VINS>

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

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

<script language="VBScript">

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

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

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


Call Msg_Log (Array("1022" , "VINS"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PAC7AJ") = Rep_JOURNAL & "\AJ"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
Call BvpEnv("VINS","PAC7BM",Rep_TMP & "\WBM.tmp")
Call BvpEnv("VINS","PAC7ES",Rep_TMP & "\WES.tmp")
Call BvpEnv("VINS","PAC7DD",Rep_USR & "\VINSDD.txt")
WshEnv("PAC7EE") = Rep_TMP & "\NULEE.tmp"
'PAC7EE not used, on default
Call BvpEnv("VINS","PAC7EQ",Rep_USR & "\VINSEQ.txt")
```

```
Call BvpEnv("VINS","PAC7ER",Rep_USR & "\VINSER.txt")
Call BvpEnv("VINS","PAC7EU",Rep_USR & "\VINSEU.txt")
Call BvpEnv("VINS","PAC7EZ",Rep_USR & "\VINSEZ.txt")
Call BvpEnv("VINS","PAC7IA",Rep_USR & "\VINSIA.txt")
Call BvpEnv("VINS","PAC7IE",Rep_USR & "\VINSIE.txt")
Call BvpEnv("VINS","PAC7IF",Rep_USR & "\VINSIF.txt")
Call BvpEnv("VINS","PAC7IG",Rep_USR & "\VINSIG.txt")
Call BvpEnv("VINS","PAC7IH",Rep_USR & "\VINSIH.txt")


If base = "ADMIN"  Then
Call BvpEnv("VINS","PAC7MA",Rep_SKEL & "\BVPMETAD")
Else
Call BvpEnv("VINS","PAC7MA",Rep_SKEL & "\BVPMETBA")
End If

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

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

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

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


</script>
</job>
```

GS/HE/ZS Files Initialization Utilities

LDGS - Introduction

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

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

On-line servers must be shut down.

None.

LDGS - Description of Steps

Initialization of the GS file: PTLDGS

This step initializes the GS file with a record.

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

LDGS - Execution Script

```
REM * ------------------------------------------------------
REM *       VISUALAGE PACBASE
REM *
REM * ------------------------------------------------------
REM *        INITIALIZATION OF THE GS FILE
REM *
REM * ------------------------------------------------------
REM *
<job id=LDGS>

<script language="VBScript">
MyProc = "LDGS"
</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" , "PTLDGS"))
'-------------------------------
WshEnv("PAC7GS") = Rep_BASE & "\GS"
Call RunCmdLog ("BVPTLDGS")

Call Err_Cod(Return , 0 , "PTLDGS")

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

</script>
</job>
```

LDHE - Introduction

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

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

Execution condition

On-line servers must be shut down.

User input

None.

LDHE - Description of Steps

Initialization of the HE file: PTLDHE

This step initializes the HE file with a record.

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

LDHE - Execution Script

```
REM * -------------------------------------------------------
REM *         VISUALAGE PACBASE
REM *
REM * -------------------------------------------------------
<job id=LDHE>

<script language="VBScript">
MyProc = "LDHE"
</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" , "PTLDHE"))
'-------------------------------
WshEnv("PAC7HE") = Rep_BASE & "\HE"
Call RunCmdLog ("BVPTLDHE")

Call Err_Cod(Return , 0 , "PTLDHE")

Call Msg_Log (Array("1023"))
```

```
'-----------------------------
Call DeleteFldr (Rep_TMP)
Wscript.Quit (Return)

</script>
</job>
```

LDZS - Introduction

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

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

Execution condition

On-line servers must be shut down.

User input

None.

LDZS - Description of Steps

Initialization of the ZS file: PTLDTS

This step initializes the ZS file with a record.

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

LDZS - Execution Script

```
REM * ------------------------------------------------------
REM *        VISUALAGE PACBASE
REM *
REM * ------------------------------------------------------
<job id=LDZS>

<script language="VBScript">
MyProc = "LDZS"
</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" , "PTLDTS"))
'-----------------------------
WshEnv("PAC7ZS") = Rep_BASE & "\ZS"
Call RunCmdLog ("BVPTLDTS")
```

```
            Call Err_Cod(Return , 0 , "PTLDTS")

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

        </script>
        </job>
```

Retrieval utilities

UTU1 UTU2 - Adaptation of 'UNS' operators

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

Principle

This tool is used to extract 'P' lines of programs containing the 'UNS' operator and without any information in the 'Level-Condition type' area. The user will have to verify the output file and delete in this file all the lines he wants to keep. For the other lines, the 'Level-Condition type' area will have to be set to '99BL' before the UTU2 procedure execution.

Execution conditions

Specify the result file name of lines to point in the startup Script: NOMUT='...'

Result

A file containing UNS 'P' lines to point 'NOMUT'.

UTU1 - Input / Processing / Results

No user input.

UTU1 - Description of Steps

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

Code	Physical name	Type	Label
PAC7PC	Save dir.: PC	Input	Sequential image of the Development Database
UTUTO	User dir.: UTUTI	Output	Lines 'P' with 'UNS' for UPDATE

UTU1 - Execution Script

```
REM * -------------------------------------------------------
REM *         VISUALAGE PACBASE
REM *
REM * -------------------------------------------------------
REM *             EXTRACTION OF LINES "P"
REM *               WITH OPERATOR "UNS"
REM *               WITHOUT LEVEL-CONDITION TYPE
REM * -------------------------------------------------------
REM *
<job id=UTU1>

<script language="VBScript">
Dim MyProc
MyProc = "UTU1"
</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" , "UTU1"))
'----------------------------------
WshEnv("PAC7PC") = BVP_SvName & ""
Call BvpEnv("UTU1","UTUTO",RepT_USR & "\UTUTI.txt")

Call RunCmdLog ("BVPTUNS1")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "UTU1")


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

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

</script>
</job>
```

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

Principle

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

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

<u>Execution conditions</u>

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

<u>Result</u>

A new sequential image of the Development Database.

UTU2 - Input / Processing / Results

No user input.

UTU2 - Description of Steps

<u>Lines 'P' update with 'UNS' operator : UTIUN2</u>

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

UTU2 - Execution Script

```
REM * -------------------------------------------------------
REM *         VISUALAGE PACBASE
REM *
REM * -------------------------------------------------------
REM *           UPDATE LINES "P"
REM *           WITH OPERATOR "UNS"
REM * -------------------------------------------------------
REM *
<job id=UTU2>

<script language="VBScript">
Dim MyProc
MyProc = "UTU2"
</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" , "UTU2"))
'----------------------------------
WshEnv("PAC7PC") = BVP_SvName & ""
Call BvpEnv("UTU2","UTUTI",RepT_USR & "\UTUTI.txt")
WshEnv("PAC7CP") = Rep_SAVE & "\PC-new"
Call RunCmdLog ("BVPTUNS2")
WshVolEnv("RC") = Return
Call Err_Cod(Return , 0 , "UTU2")


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"))
'----------------------------------
WshVolEnv("RC") = Return
Wscript.Quit (Return)

</script>
</job>
```

UTM1 UTM2 - Conversion of 'old' Meta Entities

Principle

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

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

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

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

Conversion process

Step 1: Analysis of the Database relations (UTM1)

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

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

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

The list of calls is printed in the output file.

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

You will have to specify the new relation code.

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

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

- The 'old' meta entities are converted into 'formatted' meta entities.
- The relations you have selected are modified or created.

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

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

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

    Any relation creation is performed in session 1 an in inter-library.

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

    Update

    The update is performed by the BVPACA15 program.

    A backup of the Database is recommended before the update.

    Only transactions where an error is detected are printed. It may have many rejects because these entities created in 2.n version are much more controlled than in 3.n version. And manual interventions in the Database are required.

    The transactions are not journalized.

    The &UTM2MV transaction file is declared as permanent file to allow the user to visualize all the transactions concerned by the update.

Step 3: Inventory of fixtures

It is recommended to re-execute the first step to make sure that the Database does no more contain 'old' relations calls.

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

Step 4: Reorganization

When the conversion is suitable, a reorganization of the Database is required.

Execution conditions

None during the step 1 (UTM1).

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

Printed output

At the end of the step 1, a report is printed containing the list of 'old' relations calls.

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

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

Result

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

UTM1 - Description of Steps

Analysis of the Database relations: PTUME1

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

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

UTM1 - Execution Script

```
REM * -------------------------------------------------------
REM *         VISUALAGE PACBASE
REM *
REM * -------------------------------------------------------
REM *                 CONVERSION OF OLD META-ENTITIES
REM *                 ANALYSIS OF THE DATABASE RELATIONS
REM *
REM * -------------------------------------------------------
REM * -------------------------------------------------------
<job id=UTM1>

<script language="VBScript">
Dim MyProc
MyProc = "UTM1"
</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" , "PTUME1"))
'----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
Call BvpEnv("PTUME1","PAC7EQ",Rep_USR & "\UTM1EQ.txt")
Call BvpEnv("PTUME1","PAC7ME",RepT_USR & "\UTM1ME.txt")
Call RunCmdLog ("BVPTUME1")
Call Err_Cod(Return , 0 , "PTUME1")

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


</script>
</job>
```

UTM2 - Input / Processing / Results

A '*' line in input of the second step of the conversion (UTM2) with a user
code and its password.

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

UTM2 - Description of Steps

Conversion of the meta entities, the relations and their calls: PTUME2

Code	Physical name	Type	Label
PAC7AR	Admin. Base - Base dir.: AR	Input	Development Database data
PAC7AN	Admin. Base - Base dir.: AN	Input	Development Database index
PAC7ME	User dir. : UTM1ME	Input	List of relations calls in all the entities
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PAC7MV	User dir.: UTM2MV	Output	Update transactions (length = 170)
PAC7ET	User dir.: UTM2ET	Report	Report
PAC7MB	User input	Input	user input

Update of the Administration Database: PACA15

Code	Physical name	Type	Label
PAC7AR	Base dir.: AR	Output	Development Database data
PAC7AN	Base dir.: AN	Output	Development Database index
PAC7AY	Base dir.: AY	Output	Development Database extension
PAC7AJ	NUL	Output	Development Database Journal
PAC7AE	System - Skel. dir.: AE	Input	Error messages
PACGGN	Admin. Base - Base dir.: AN	Input	Administration Database index
PACGGR	Admin. Base - Base dir.: AR	Input	Administration Database data
PACGGY	Admin. Base - Base dir.: AY	Input	Administration Database extension
PACGGU	Admin. Base - Base dir.: GU	Input	Administration Database users
PAC7DC	NUL	Input	DSMS elements file of the Development Database
PAC7ME	NUL	Input	Work file
PAC7MV	User dir.: UTM2MV	Input	Update transactions

Code	Physical name	Type	Label
PAC7RB	NUL	Output	UPDT erroneous transactions (length=80)
PAC7RY	NUL	Output	UPDP erroneous transactions (length=310)
PAC7IE		Report	Update report (length=132)
PAC7IF		Report	List of erroneous transactions (length=132)

The list of transactions belonging to a user is preceded by a banner specifying the user code.

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

UTM2 - Execution Script

```
REM * -------------------------------------------------------
REM *        VISUALAGE PACBASE
REM *
REM * -------------------------------------------------------
REM *               CONVERSION OF OLD META-ENTITIES
REM *               THE RELATIONS AND THEIR CALLS
REM *
REM * -------------------------------------------------------
REM * -------------------------------------------------------
<job id=UTM2>

<script language="VBScript">
Dim MyProc
MyProc = "UTM2"
</script>

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

<script language="VBScript">

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

NoDisp = "NO"

Call Msg_Log (Array("1022" , "PTUME2"))
'--------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AN") = Rep_BASE & "\AN"
```

```
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7MB") = Fic_Input
Call BvpEnv("PTUME2","PAC7ET",Rep_USR & "\UTM2ET.txt")
Call BvpEnv("PTUME2","PAC7ME",RepT_USR & "\UTM1ME.txt")
Call BvpEnv("PTUME2","PAC7MV",Rep_USR & "\UTM2MV.txt")
Call RunCmdLog ("BVPTUME2")
Call Err_Cod(Return , 0 , "PTUME2")


Call Msg_Log (Array("1022" , "PACA15"))
'-----------------------------------
WshEnv("PAC7AE") = Rep_SKEL & "\AE"
WshEnv("PAC7AJ") = Rep_TMP & "\NULAJ.tmp"
'PAC7AJ not used, on default
WshEnv("PAC7AN") = Rep_BASE & "\AN"
WshEnv("PAC7AR") = Rep_BASE & "\AR"
WshEnv("PAC7AY") = Rep_BASE & "\AY"
WshEnv("PACGGN") = Rep_ABASE & "\AN"
WshEnv("PACGGR") = Rep_ABASE & "\AR"
WshEnv("PACGGU") = Rep_ABASE & "\GU"
WshEnv("PACGGY") = Rep_ABASE & "\AY"
WshEnv("SEMLOCK") = Rep_BASE & "\LO"
WshEnv("SEMADMIN") = Rep_ABASE & "\LO"
WshEnv("PAC7IE") = Rep_TMP & "\NULIE.tmp"
'PAC7IE not used, on default
Call BvpEnv("PACA15","PAC7IF",Rep_USR & "\UTM2IFA15.txt")
Call BvpEnv("PACA15","PAC7MV",Rep_USR & "\UTM2MV.txt")
WshEnv("PAC7ME") = Rep_TMP & "\NUL.tmp"
'PAC7ME not used, on default
WshEnv("PAC7DC") = Rep_TMP & "\NULDC.tmp"
'PAC7DC not used, on default
WshEnv("PAC7RB") = Rep_TMP & "\NULRB.tmp"
'PAC7RB not used, on default
WshEnv("PAC7RY") = Rep_TMP & "\NULRY.tmp"
'PAC7RY not used, on default
Call RunCmdLog ("BVPACA15")
If Return = 4 Then
Call Msg_Log (Array("1060"))
End If
Call Err_Cod(Return , 4 , "PACA15")

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

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


</script>
</job>
```

IBM ®

Part Number: DELNT001354A - 6889

Printed in USA